Multimodal Spatiotemporal Collaborative Positioning Framework for Indoor Environments

Sakr, Mostafa

UNIVERSITY OF CALGARY

Multimodal Spatiotemporal Collaborative Positioning Framework for Indoor Environments

by

Mostafa Sakr

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN GEOMATICS ENGINEERING

CALGARY, ALBERTA

JULY, 2019

Abstract

This thesis proposes and evaluates a unified collaborative and multimodal framework for indoor positioning and mapping using smartphones. The proposed framework aims to harness the potential of collaboration between different nodes for the positioning and mapping tasks, using only smartphones, without assuming the existence of any specific infrastructure. This objective is achieved by first exploring and enhancing the different building blocks of the proposed framework; followed by evaluating the accuracy gains from using a collaborative approach to the positioning problem.

The first building block to be studied is the standalone navigation filter. The standard extended Kalman filter, the unscented Kalman filter, and the particle filter were evaluated for node positioning using the pedestrian dead reckoning model as a system model, while the measurement update is achieved using Wi-Fi fingerprinting with a Gaussian process model.

The second component of the system is the Wi-Fi radio map. The proposed framework utilizes a new sparse Gaussian process model to represents the Wi-Fi radio map, used for Wi-Fi signal strength-based fingerprinting. The map building algorithm using the proposed model and its performance are presented and discussed.

The collaboration between different nodes is examined in detail, and a new family of distributed particle filters for collaborative positioning applications are introduced. The detailed derivation of the filtering equation along with simulation evaluation of the filters are presented.

The collaboration model used in the proposed framework is based on the relative range measurements. A ranging device based on ultra-wideband (UWB) technology is designed and implemented to evaluate the framework. The ranging device is based on the DW1000 UWB transceiver from Decawave. The device can reach centimetre-level ranging accuracy and

connects to a host microcontroller which controls the flow of ranging messages, computes the range, and communicate with a paired smartphone through Bluetooth Low Energy interface. On the smartphone, a logging application saves the range information from the UWB device along with other sensors data such as accelerometer, gyroscope, magnetometer, pressure, and Wi-Fi signal strength. Along with this software, a simulation environment is developed to model the motion of random nodes inside an indoor environment. This simulator was used in the evaluation of the proposed particle filters family. The thesis concludes by evaluating the proposed framework using multiple test trajectories and different operating scenarios in a challenging indoor environment.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor, Prof. Naser El-Sheimy. This thesis would not have been possible without his continuous support and guidance throughout my research. Thank you for the opportunity to be part of the Mobile Multi-Sensor Systems (MMSS) research group, for your patience, and for the freedom to conduct my research.

I want to thank Andrea Masiero from the interdepartmental research Centre of Geomatics (CIRGEO) at the University of Padua, Padua, Italy, for his invaluable help in the experimental part of this work, and his insightful comments and feedback.

I want to thank my current and former colleagues in the MMSS group: Adel Moussa for the informative and inspiring discussions; Zahra Lari for her help and feedback during the early stage of my research; Shady Zahran, Mostafa Mostafa, and Mohamed Hassanein for helping in the experimental part of this work; Naif Alsubaie, Elyar Sabbaghian, Ahmed Radi, Maan Khedr, Chunyang Yu, Qifan Zhou, You Li, Abdulla Al-Rawabdeh, Haiyu Lan, Hussien Sahli, Hani Mohammed, Abanob Salib, and Yiran Luo for the interesting discussions and comments.

During my studies, I was fortunate to work with a group of remarkable individuals in the Geomatics Graduate Group (G^3). I would like to thank Yan Xiang, Yuting Gao, Chandra Tjhai, Eric Wang, Sandra Simeonova, Asal Naghdi, Rodrigo Silva, and Paul Gakne.

I would like to thank June Au Yeung and Monica Freeman from the Department of Geomatics Engineering for their help and dedication.

Last but not least, I would like to thank my wife, Shaimaa, and my kids, Yassin and Sophia, for their support and understanding; my sister, Shaimaa, for her sense of humour and encouragement that make my life brighter; and my parents, Mahmoud and Ehsan, for their unwavering support, dedication, and care over the years, without which none of this would have been possible.

Dedication

To my father Mahmoud Sakr and my mother Ehsan Hamdy.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Definition
5G	5^{th} generation
AOA	Angle of arrival
AP	Access point
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BLE	Bluetooth low-energy
CDF	Cumulative density function
CRB	Cramér-Rao bound
CRF	Conditional random field
CRLB	Cramér-Rao lower bound
CSI	Channel state information
CUPT	Constant velocity update
DAG	Directed acyclic graph
DR	Data rate
DTC	Deterministic training conditional
EFIM	Equivalent Fisher information matrix
EKF	Extended Kalman filter
EM	Expectation maximization
FCC	Federal communications commission
FIM	Fisher information matrix
FITC	Fully independent training conditional
FM	Frequency modulation
GATT	Generic attribute profile
GDOP	Geometric dilution-of-precision
GMRK	Gaussian Markov random field
GNSS	Global navigation satellite system
GP	Gaussian process
GSM	Global system for mobile communications
IMU	Inertial measurement unit
INS	Inertial navigation system
IRLS	Iterative re-weighted least squares
KLD	Kullback-Leibler divergence
KNN	K-Nearest Neighbour
LBFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno
LBS	Location-based services
LiDAR	Light detection and ranging
LS	Least-squares
LVM	Latent variable model
MAC	Medium access control
MAP	Maximum a posteriori
MCU	Microcontroller
ML	Maximum-likelihood
MMSE	Minimum mean-squares error

mmWave	Millimetre waves
NA	Network adjustment
PAN	Personal area network
PDF	Probability density function
PDR	Pedestrian dead-reckoning
PF	Particle filter
PGSGP	Parametric grid sparse Gaussian process
PLEN	Preamble length
PLM	Path-loss-model
PRF	Pulse repetition frequency
RBCPF	Rao-blackwellized collaborative particle filter
RBF	Radial basis function
RBPF	Rao-blackwellized particle filter
RF	Radio-frequency
RFID	Radio-frequency identification
RGCPF	Rao-blackwellized Gaussian-importance particle filter
RMS	Root-mean-square
RMSE	Root-mean-square error
RSS	Received signal strength
RSSI	Received signal strength indicator
RTLS	Real-time location systems
SIR	Sample importance resampling
SKF	Schmidt-Kalman filter
SLAM	Simultaneous localization and mapping
SNR	Signal-to-noise ratio
SPI	Serial peripheral interface
SR1	Symmetric rank one
SVN	Support vector machine
TDMA	Time-division-multiple-access
TDOA	Time difference of arrival
TOA	Time of arrival
TOF	Time-of-flight
UKF	Unscented Kalman filter
UWB	Ultra-wideband
VHF	Very high frequency
Wi-Fi	Wireless fidelity
WSN	Wireless sensor network
ZUPT	Zero-velocity update

Chapter One: **Introduction**

The importance of indoor positioning and navigation is accentuated by two major technological trends that persisted for the last few decades. First, the proliferation of smart devices in all aspects of life and the continuous growth of its computational and sensing capabilities. And, second, the incessant increase in connectivity and bandwidth available to these devices, which was a crucial driver for the development of the next generation of mobile networks (5G) [1]. This is manifested by the expected growth of the location-based services (LBS) and the real-time location systems (RTLS) market value to US$ 68.85 billion by 2023 [2].

Indoor positioning frameworks and algorithms are essential components to enable location-based services and real-time location systems. The commercial applications of LBS and RTLS include mobile advertising, navigation, tourism, local search, and location-based health information. The LBS and the RTLS can also be used for in an industrial context, to monitor inventory and assets, to control access to certain areas, and to ensure the safety of workers in hazardous situations. Additionally, indoor positioning and navigation have many applications in the military and law-enforcement domains.

The field of indoor navigation and positioning continues to be a very active research area, though the emphasis of the research community has shifted towards increasing accuracy, availability, and reliability of the indoor positioning solution, while in the same time addressing the practical aspects of system deployment and decreasing the cost and time of system training. The advances in indoor navigations and positioning field is highlighted in many survey papers and monographs, covering different aspects of the problem, see [3]–[14].

Indoor positioning algorithms can be categorized according to the type of the signal or the information used as an input to the algorithm [13]. Different signals can be used for indoor

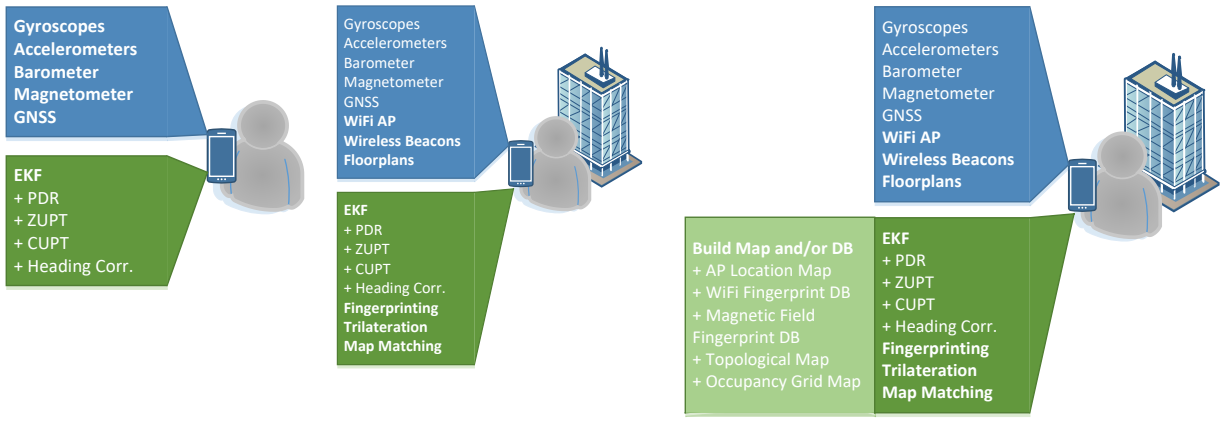
navigation, including, but not limited to, Wi-Fi, Bluetooth, RFID, mmWave, acoustic signal, image, visible light, smartphone dead-reckoning, magnetic field, ultra-wideband (UWB), FM signals, and infrared. Another classification of the indoor positioning systems depends on the type of physical measurement inferred from the observed signal [15]. The different measurements include: distance measurement using the signal strength, the time of arrival, or the time difference of arrival; direction of arrival; area measurements, in which the signal coverage region is described by a geometric shape; hop count between different nodes, where the connection between nodes is limited by the maximum range of their radio; neighbourhood proximity measurements.

Another possible classification of the navigation algorithms considers the type of the observed quantity used for navigation [16]:

1. Proprioceptive: by observing quantities internal to the navigating nodes, using gyroscopes and accelerometers, the position of a mobile node can be determined by either using the classical form of inertial mechanization equations or by using pedestrian dead-reckoning (PDR). As such, this class of observations provides direct observations of the orientation and position states of the mobile nodes or its derivatives.
2. Exteroceptive: by observing external environmental features, such as floorplan features, Wi-Fi access points locations or signal levels, external magnetic field intensity, or distance from landmarks, the state of the mobile node is indirectly inferred from the observed environmental features.

By considering the previous two classes, it would be clear that using internal information only will not involve any knowledge of the surrounding environment. The only requirement for a positioning system is to define a navigation frame of reference–either local or global–along with

an initial value for the position state and its derivatives, according to the order of the model. However, the accuracy of the position solution derived using this form of dead-reckoning degrades with time, due to the accumulation of sensor errors. As a result, the meter-level positioning accuracy is not attainable by this approach alone. One approach to mitigate this problem is to make use of the environmental features to enhance the overall accuracy of the positioning solution.



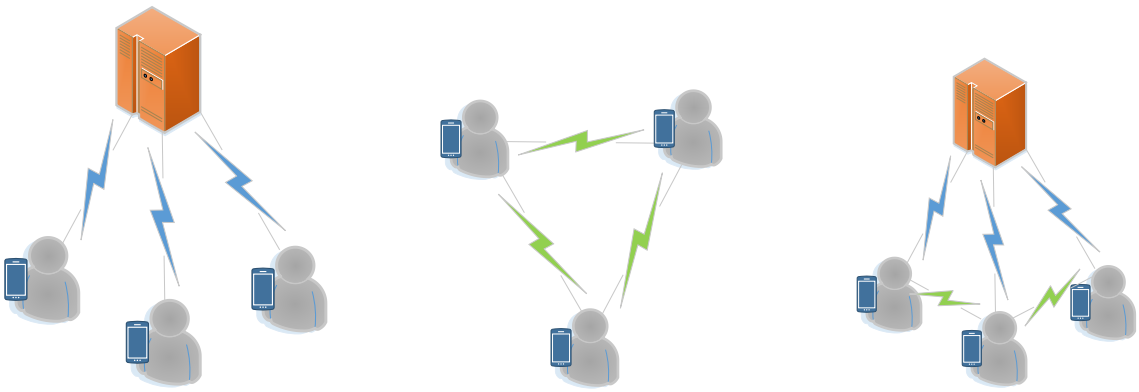
(a) (b) (c)

Figure 1–1 Different configuration of standalone navigation; (a) proprioceptive sensors, (b) proprioceptive and exteroceptive sensors, (c) proprioceptive and exteroceptive sensors, with autonomous maps creation

In addition to the internal state observation, provided by inertial sensors, for example, the mobile nodes can observe a multitude of environmental features. These observations are then fused with the internal observations, to limit the state drift. In order to use these features, they should be associated with a specific physical location within the navigation frame, through a map. Then, this information can be used to infer the position of the mobile node indirectly.

However, building this kind of maps is both cost- and labour- intensive. Much research is directed towards automating and increasing the efficiency of map generation. The research in this category can be classified into two schemes [10]:

1. Explicit crowdsourcing approaches. The users are consciously collecting data, and tagging the location of the observed environmental features, or landmarks, observed by the mobile node. Then, this data is aggregated into databases or maps and shared among the different users of the system.
2. Implicit crowdsourcing approaches. The users are unaware of the data collection and database or map building process. The data collection process runs in the background, while users are navigating through the area of interest.



(a) (b) (c)

Figure 1–2 Collaboration configuration for navigation; (a) centralized approach, (b) distributed approach, (c) hybrid approach

This classification of standalone navigation according to the type of used sensors and features, is summarized in Figure 1–1. Figure 1–1 (a) shows a standalone navigation scheme utilizing only proprioceptive sensors, along with an example of the used algorithms to calculate the position of the mobile node. Figure 1–1 (b) shows a navigation scheme utilizing both proprioceptive and

exteroceptive sensors, with pre-built maps of the environment. Finally, Figure 1–1 (c) shows the same configuration but with the autonomous map creation scheme.

Another classification for the navigation algorithms is concerned with whether the positioning and navigation algorithm is collaborative or carried out by each mobile node individually.

In collaborative approaches, the position of each mobile node is calculated, taking into consideration the positions and uncertainties of a subset of the collaborating nodes. In this scheme, a method of evaluating the relative position between different nodes is required.

In non-collaborative approaches, the position of each node is considered separately, without taking into consideration its relation to the surrounding nodes.

Figure 1–2 summarizes the different approaches for collaborative navigation, which are described as follows:

1. A centralized approach, where the state estimates for all participating nodes are calculated in a centralized remote computer. In this approach, all participating nodes send their observations to the remote central node, and then they receive their estimated states from that remote node. This approach is only feasible when high bandwidth data links are available, the amount of transmitted data is low, or the number of collaborating nodes is relatively small.
2. A distributed approach, in which each node is responsible for its state estimation. Though, each node also shares the estimate, the observations, or a combination of them with some of the neighbouring nodes. This process can be performed once, or repeated for several times until the entire population of mobile nodes reaches a consensus.
3. A hybrid approach, in which the mobile nodes share a subset of their state estimate to enhance the overall positioning accuracy of the collaborating nodes. Additionally, the

different nodes can communicate with a centralized node, to exchange map information and to facilitate peer discovery and communication.

Any collaborative positioning framework involves several components such as the standalone navigation algorithm, the collaboration methods, and the overall positioning framework. In order to design and evaluate a collaboration scheme between multiple nodes, the following questions should be examined:

1. How to calculate/update the state of each node, regardless of the type of collaboration adopted in the system? Specifically, what is the definition of the node internal states, what is the type of observables, and what is the relation between the observables and the internal nodes?
2. What is the collaboration type between nodes? How collaboration occurs between different nodes? Moreover, what kind of data is shared among the collaborating nodes? Do the participating nodes share their full states and the raw measurements every time epoch?
3. Where to compute the updated states of the collaborating node? Does this process take place inside each node individually? Alternatively, does it take place inside a centralized processing center, then it is distributed to the collaborating nodes?
4. What environmental features to map, and how to represent and build the maps? Moreover, how to disseminate the maps between the individual nodes, and how incorporate them in a globally shared map?

This thesis will attempt to answer these questions in the context of indoor positioning and navigation problem, to conceive a unified framework for indoor positioning for mobile nodes equipped with smartphone devices. The rest of this chapter will try to elaborate on the current

state-of-the-art answers for the previous questions. The following section will discuss the motivations and objectives of this thesis. Finally, the outline of this thesis is presented.

1.1 Background

The section starts by reviewing the basic concepts of wireless positioning. Then, it reviews the recent advances in Wi-Fi positioning and localization as an answer to the single node localization, with a focus on fingerprinting methods. The following subsection reviews recent and prominent contributions in collaborative indoor navigation. It is worth noting that the fingerprinting techniques are not unique to Wi-Fi signals, and it can be applied to other types of signals, such as magnetic fields [17]–[20]. Finally, using simultaneous localization and mapping (SLAM) algorithms for indoor navigation, to mitigate the laborious and expensive process of mapping the environmental features, is discussed, and the recent advances are reviewed.

1.1.1 Wireless Positioning Concepts

Radio signals have been used as a tool to measure distances between transmitters and receivers since the early days of wireless communication [21]–[23]. The use of radio signals for indoor positioning started to gain traction more recently, with the evolution of the wireless communication technologies, the continuous improvements in microelectronics, and the wide adoption of personal and portable communication devices.

Early indoor localization systems used custom-built hardware for identifying users. Such legacy systems used technologies ranging from infrared-based proximity detection, ultrasound-based trilateration, and magnetic-field tracking. They also explored using Wi-Fi or vision technology for localization [24]. More recently, the focus has shifted to Wi-Fi positioning and localization, the use of wireless fingerprinting techniques, and the online fingerprinting technologies [25], [10], [12], [11], [14].

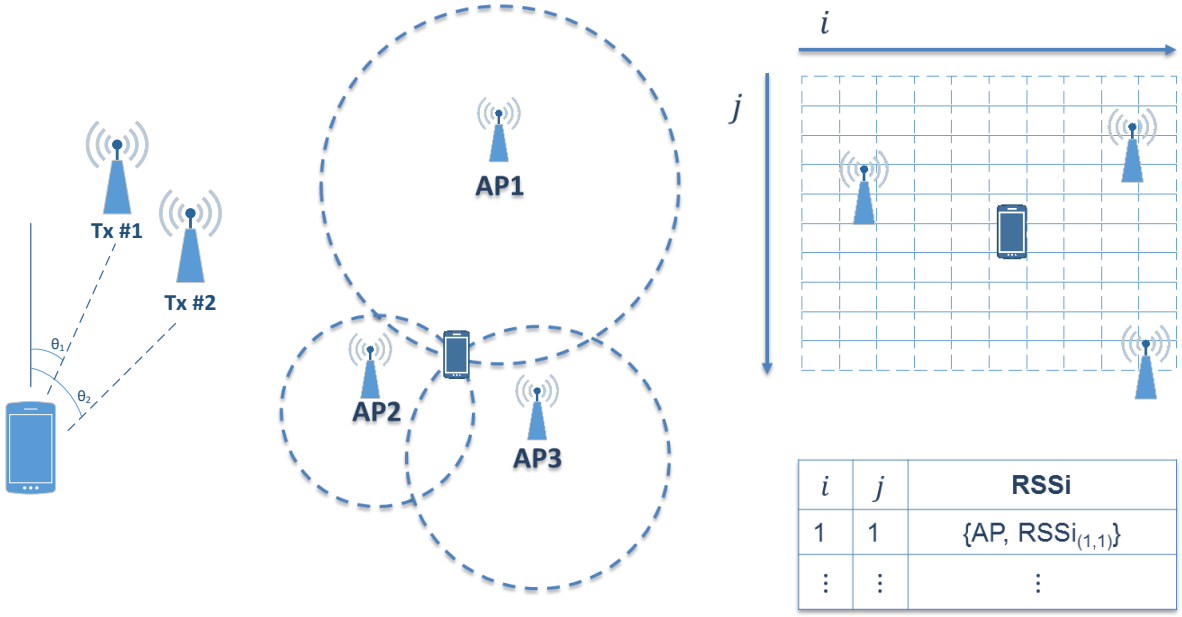
Figure 1–3 summarizes the main methods of wireless positioning. The distinction between each category is based on the type of information derived from the wireless signal, which dictates a specific receiver and antenna configuration. In brief, the use of wireless signals for localization and positioning can fall under one of the following three categories [25]:

1. *Triangulation*: using the wireless signal Angle of Arrival (AOA). This method can be implemented using an antenna array on the receiver side, and the phase difference between different antenna indicates the direction to the transmitter.
2. *Trilateration*: which can be further divided according to the specific property of the wireless signal that is used to infer distance:
    a. Time of Arrival (TOA): in this approach, the transmitter and receiver should be synchronized, and the signal travel time is recorded. The travel time can be converted into the distance by multiplying the time by the signal propagation speed.
    b. Received Signal Strength (RSS): by measuring the received signal strength, the distance between the transmitter and receiver can be inferred, by the use of the log-distance path loss model from radio signal propagation [26]:

$$P_r(d) = P_0 - 10 \cdot n \cdot \log\left(\frac{d}{d_0}\right) + \epsilon_r \tag{1.1}$$

    $P_r(d)$ is the received signal power received at distance d, P_0 is the power at a reference distance d_0, n is the power loss exponent, and $n = 2$ for free space, and it ranges from 1.6 to 3.3 for different types of indoor environments (ibid., Table 2), and ϵ_r is a zero-mean Gaussian random variable, in dB units.
3. *Fingerprinting*: this technique usually utilizes the RSS values–or the channel state information (CSI) [27], [28]–from the different Wi-Fi access points. It is divided into two

steps; offline site surveying, and online positioning. In the offline stage, the RSS values and the associated physical locations are recorded. Then, in the online stage, the mobile node collects RSS signals and compare it with the pre-surveyed map, to determine its current location. More information about this technique is provided in the following section.



(a) (b) (c)

Figure 1–3 Wireless positioning basics; (a) triangulation, (b) trilateration, (c) RSS fingerprinting

The trilateration and triangulation methods require the knowledge of the precise locations of the access points, or the wireless transmitters in general, to be able to localize the mobile in the mapping frame. On the other hand, fingerprinting techniques require accurate maps of the environmental features used–for example, Wi-Fi RSS signal, radio channel CSI, or magnetic field intensity. These maps are usually hard to obtain and maintain, and building them from scratch is a labour- and time-intensive task–possible solutions to this problem are discussed in

Sections 1.11.1.3 and 1.1.4. Additionally, trilateration and triangulation methods are sensitive to multipath effects and require a direct path between the receiver and the wireless transmitter. This line-of-sight requirement is dropped for the fingerprinting techniques.

1.1.2 Wi-Fi Fingerprinting Overview

Using Wi-Fi fingerprinting for positioning is typically divided into two stages: an offline pre-surveying phase, and an online positioning phase [18], [29]. In the first stage, the Wi-Fi fingerprints database, or map, is built as tuples of the Wi-Fi RSS vector from all visible access points (AP), along with the location of the observation:

$$FP_i = \{X_i, (MAC_{i,1}, RSS_{i,1}), (MAC_{i,2}, RSS_{i,2}), \dots, (MAC_{i,N}, RSS_{i,N})\}, \quad (1.2)$$

where, FP_i is the i-th entry in the fingerprints database, N is the total number of observed access points at the location X_i, and $(MAC_{i,j}, RSS_{i,j})$ are the j-th AP MAC address and RSS value, correspondingly. In the online positioning phase, the location of a moving node can be determined by comparing the current Wi-Fi RSS reading against the pre-built fingerprint database, using different methods [18], [29]–[31]. The Wi-Fi fingerprinting systems can fall under one of the following categories; deterministic [32] and probabilistic [33], summarized in Table 1–1 and discussed briefly in the following subsections.

Table 1–1 Summary of Wi-Fi Positioning Algorithms

Type	Positioning Algorithm	Framework
Deterministic	Distance Minimization	Nearest Neighbour
	(Weighted) Average	K-Nearest Neighbour
	Binary Classification	Support Vector Machine
Probabilistic	Posterior Expectation	Bayesian Network
	Maximum a posteriori	Naïve Bayes
	Minimize KLD	Kullback-Leibler Divergence
	Maximum a Posteriori (Viterbi)	Conditional Random Field
	Particle Filter	Gaussian Process
	Maximum a Posteriori	Gauss Markov Random Field

1.1.2.1 Deterministic Methods

The deterministic methods determine the distance between the measurements vector and the training data, and use the position of the nearest point, or points, to estimate the position of the mobile node. The main difference between the different deterministic techniques is the method of calculating the distance between the measurements vector and the training data. A summary of several distance measures is presented in Table 1–2.

Table 1–2 Summary of common distance measures

Euclidian distance (2-norm)	$D_{L^2} = \sqrt{\sum_{i=1}^{N}(r_i - z_i)^2}$
Manhattan distance (1-norm)	$D_{L^1} = \sum_{i=1}^{N}\lvert r_i - z_i\rvert$
Infinity-Norm	$D_{L^\infty} = \max_i(\lvert r_i - z_i\rvert)$
Mahalanobis distance	$D_M = \sqrt{(R-Z)^T W (R-Z)}$

Where, z_i represents the RSS measurement from the i-th access point, r_i is the reference RSS for the i-th access point, N is the total number of access points included in the current observation, $Z = \{z_1, \dots, z_N\}$, $R = \{r_1, \dots, r_N\}$, W is a $N \times N$ weighting matrix, and D is the measured distance.

1.1.2.1.1 Nearest Neighbour Methods

Nearest neighbour is one of the simplest methods to find the position of a node, in which the distance between the measurements vector and the training data is calculated. Then, positioning is achieved by finding the index of the fingerprint database entry that minimizes the calculated

distance. The position stored in that entry is assigned to the mobile node, $\hat{x}$, as shown in Equation (1.3).

$$\hat{x} = \arg\min_{x}[D] \tag{1.3}$$

The algorithm calculates the distance, D, and find the position of the entry, x, which minimizes the selected distance measure.

1.1.2.1.2 K-Nearest Neighbours Method

K-nearest neighbours (KNN) method is similar to the nearest neighbour method; however, instead of locating one nearest position, the average of the K nearest points are used to estimate the position of the mobile node [18], [34], [35]. The algorithm starts by calculating the distances using one of the distance measures; then, it uses Equation (1.3) to determine the K nearest points. The position of the mobile node is the average of the positions of the nearby training points, x_i:

$$\hat{x} = \frac{1}{k}\sum_{i=1}^{k} x_i \tag{1.4}$$

Another approach is to calculate the weighted k-nearest neighbour method [35], given by:

$$\begin{cases} \hat{x} = \dfrac{1}{k}\displaystyle\sum_{i=1}^{k} w_i \cdot x_i \\ \displaystyle\sum_{i=1}^{k} w_i = 1 \end{cases} \tag{1.5}$$

Where, w_i, is the weight given to each point of the k-nearest points.

1.1.2.1.3 Support Vector Machine

Support vector machine (SVM) can be used to determine whether a particular RSS vector belongs to a specific database entry or not [36]; i.e., a binary classification problem over each database entry. The training phase of the SVM involves finding a hyperplane that separates the

dataset into two regions while maximizing the margin between the boundary and the training dataset.

The hyperplane equation is given by:

$$w^T x + b = 0 \tag{1.6}$$

where x is the RSS vector, and w is the normal vector to the hyperplane. The optimization goal is to maximize the separation between the hyperplane and closest training data point i:

$$\frac{|w^T x_i + b|}{\|w\|} \tag{1.7}$$

This maximization can be expressed as minimization of $\|w\|$, the Euclidian norm of the vector w, subjected to the condition [37]:

$$y_i(w^T x_i + b) \geq 1 \; \forall i \tag{1.8}$$

where, y_i is the training data label. This condition ensures that there are no training points in the gap separating the two classification regions.

1.1.2.2 Probabilistic Methods

This section introduces the general idea of the probabilistic approaches of positioning; then, it will highlight several probabilistic frameworks targeting positioning using Wi-Fi signal strength measurements. In probabilistic methods, the position of the mobile node is treated as a random variable, and the objective of the positioning algorithm is to find the posterior probability distribution of such variable, given an RSS measurements vector, Z, and the pre-surveying map, FP. The position estimate could be the value that maximizes the posterior probability distribution, Equation (1.9):

$$\hat{x} = \arg\max_x [p(x|Z, FP)] \tag{1.9}$$

Alternatively, the position estimate could be the mean of the posterior probability distribution [30], as shown in Equation (1.10):

$$\hat{x} = \int x \cdot p(x|Z, FP)\,\mathrm{d}x \tag{1.10}$$

In order to estimate the position, the posterior probability distribution function has to be evaluated given the current measurements, $p(x|Z)$. Using Bayes rule, the posterior distribution can be represented by Equation (1.11):

$$p(x|Z) = \frac{p(Z|x) \cdot p(x)}{p(Z)} = \frac{p(Z|x) \cdot p(x)}{\sum_{x'} p(Z|x') \cdot p(x')} \tag{1.11}$$

where, $p(Z|x)$ is the likelihood of the measurement at position x, and $p(x)$ is the prior distribution of the node position.

Probabilistic methods can be categorized according to the description of the likelihood function. Additionally, they can be divided into static positioning approaches and recursive estimation or filtering approaches [38]. In the static positioning case, the estimate is not delay-sensitive, and the algorithm estimates the entire trajectory when all measurements are available. In the dynamic positioning case, the position estimate is updated in real-time with each newly available measurement. In this approach, the update is performed using a filtering algorithm, such as Kalman filtering or particle filtering.

1.1.2.2.1 Bayesian Networks

A Bayesian network is a directed acyclic graph (DAG), that represents a joint probability distribution between different random variables, along with the dependency between these variables [39]. The use of Bayesian networks for positioning, based on Wi-Fi signal strength, was first proposed by [40], where the Bayesian network is used to infer the location of a mobile node, based on the signal-to-noise ratio (SNR) values received from different access points. The

Bayesian network is used to evaluate the probability of being at a position x given the observations vector Z. In this approach, the distinguishable locations and measurements values are discrete.

The likelihood function can take many forms, for example, a kernel form–such as Gaussian kernel–or a histogram form [30], [41]. The histogram representation of a probability distribution is built by dividing the possible variable range into a set of non-overlapping bins, resulting in a piecewise constant distribution function with k parameters, where k is the number of bins. The posterior over the location conditioned on the signal strength measurement, $p(x|Z)$, is calculated using Bayes rule assuming a uniform prior distribution, and the location is calculated as the expectation of the posterior distribution–if the location variable is numeric–i.e., $\mathbb{E}[x|Z] = \sum_{x_i \in \mathcal{X}} x_i \cdot P(x_i|Z)$.

1.1.2.2.2 Naïve Bayes

The naïve Bayes assumption is used to simplify the evaluation of the joint probability distribution during the offline training phase [42]. The naïve Bayes assumption postulates that the observations are independent. As a result, instead of evaluating the joint probability distribution of the observations directly, $p(z_1, \dots, z_n)$, which requires a lot of training data, the joint probability is evaluated as the product of individual probability density functions of the different observations, $\prod_{i=1}^{n} p(z_i)$, where z_i is the signal strength observation from the access point i, and n is the total number of the access points observed at a certain location. Each distribution $p(z_i)$ is evaluated as a normalized histogram. The online positioning is achieved by finding the location that maximizes the posterior distribution. Additionally, multiple samples can

be used to increase the accuracy of location estimation. The locations are clustered according to the common access points observed, to reduce the required search space.

1.1.2.2.3 Kullback-Leibler Divergence

The positioning problem of a mobile node, using signal strength measurements, can be posed as a hypothesis testing problem [43]–[46]. This stochastic framework depends on several stationary sensors in the environment, called *cluster heads*, which receives the transmitted signal by the mobile nodes themselves. Although this configuration is not the one typically used in Wi-Fi positioning techniques–it is more convenient in the wireless sensors network context–the theoretical approach is interesting for the usual case, in which the stationary points, i.e., access points, are transmitting signals, while the mobile nodes are measuring the signal strength values of the received signals. In this approach the cluster heads collect a sequence of observations from the mobile nodes and based on the result of the hypothesis test, it can assign the mobile nodes to one of a discretized set of possible locations.

A similar approach is applied to solve the positioning problem using Wi-Fi signal strength measurements [47]. In which the statistical signature generated during run-time is compared to the statistical signatures of each cell. These signatures are generated during the training phase. The criteria used to quantify the separation between different probability density functions, corresponding to the signatures, is the Kullback-Leibler Divergence (KLD), which is defined as:

$$KL(p_r||q_i) = \sum p_r(s) \ln \frac{p_r(s)}{q_i(s)} \tag{1.12}$$

where, $p_r(s)$ is the run-time signature distribution, and $q_i(s)$ is the training distribution at cell i. The summation in Equation (1.12) is performed over all the pins in the distribution.

The distribution of the signature is a multivariate Gaussian distribution, given by:

$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{K/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \quad (1.13)$$

The Kullback-Leibler Divergence of this distribution is given by:

$$KL(p_r||q_i) = \frac{1}{2}((\mu_i-\mu_r)^T\Sigma_i^{-1}(\mu_i-\mu_r) + \text{tr}\,(\Sigma_r(\Sigma_i)^{-1} - I) - \ln|\Sigma_r(\Sigma_i)^{-1}|) \quad (1.14)$$

The location of the mobile node is the coordinates of the cell which minimize $KL(p_r||q_i)$.

1.1.2.2.4 Conditional Random Field

Conditional random field (CRF) [48] is used for localization using RSS measurements [49]. The conditional random field is an undirected probabilistic model that captures the conditional distribution $p(x|Z)$, without considering he marginal probability, $p(Z)$. The main attraction of CRF models is the elimination of the need to model the dependence between the observations, without using the independence assumption–unlike the Naïve Bayes models. The proposed system depends on building a graph that represents all reachable locations over a given floorplan, thus capturing the physical constraints on the map. The states and transitions are constrained using multiple feature functions, f_i, which are combined to form a potential function, Ψ, defined by the Equation (1.15):

$$\Psi(S_{j-1}, S_j, Z, j) = \exp\left(\sum_{i=1}^{m} \lambda_i \times f_i(S_{j-1}, S_j, Z, j)\right) \quad (1.15)$$

The conditional probability $p_\lambda^*(S|Z)$ is defined in terms of the potential functions, as shown in Equation (1.16):

$$p_\lambda^*(S|Z) \propto \prod_{j=1}^{T} \Psi(S_{j-1}, S_j, Z, j) \quad (1.16)$$

where $S = \{S_0, \dots, S_T\}$ is a sequence of the state variables representing the location of a mobile node, $Z = \{Z_0, \dots, Z_T\}$ is a sequence of observations, and the subscript $\{0, \dots, T\}$ indicates the time, and λ_i is a weighting factor determined during the training stage. The inference or localization stage of this approach is achieved by finding a trajectory S^* that maximizes the conditional probability $p(S|Z)$:

$$S^* = \arg\max_{S} p(S|Z) \tag{1.17}$$

The solution to this optimization problem is achieved using the Viterbi algorithm.

1.1.2.2.5 Neural Networks and Deep Learning

A neural network is a non-linear statistical model that can be used for regression and classification [50]. In its purest forms, a neural network is composed of P inputs, K outputs, and it has one hidden layer containing M nodes connecting the inputs to the outputs. This simple network is fully described by Equation (1.18).

$$\begin{aligned} Z_m &= \sigma(\alpha_{0m} + \alpha_m^T X), \quad && m = 1, \dots, M, \\ T_k &= \beta_{0m} + \beta_k^T Z, && k = 1, \dots, K, \\ Y_k &= f_k(X) = g_k(T), && k = 1, \dots, K \end{aligned} \tag{1.18}$$

where $X = (X_1, \dots, X_M)$, $Z = (Z_1, \dots, Z_M)$, $T = (T_1, \dots, T_K)$, $\sigma(\cdot)$ is the activation function, $g_k(T)$ is the output function, and $\{\alpha_m, \beta_m\}$ are a set of weights.

A typical selection for the activation function is the sigmoid function, shown in Equation (1.19):

$$\sigma(v) = \frac{1}{1 + e^{-v}} \tag{1.19}$$

The objective of the learning process of the model is to find the values of the weights that fit the training data by minimizing a cost function using the back-propagation approach.

Artificial neural networks have been used for indoor positioning using Wi-Fi or Bluetooth fingerprinting [51]–[55]. An earlier approach used the artificial neural networks to learn the grid-based and topological map of an indoor environment [56]. Recently, the resurgence of the deep learning algorithms has renewed the interest in using neural networks for indoor positioning. Deep neural networks have been used for channel state information (CSI) fingerprinting [28]. It has been used for Wi-Fi positioning and fingerprinting [57]–[60].

1.1.2.2.6 Gaussian processes and Gaussian Markov random field

Gaussian processes (GP) [61], or Gaussian random field and Gaussian Markov random field (GMRF) [62] are used for mobile nodes localization using RSS measurements. The Gaussian Process model is a non-parametric model [63], which means it depends on training data in order to predict the RSS value at any given point. The training data set is composed of a trace of Wi-Fi signal strength values annotated by the location. The Gaussian process can be approximated as a Gaussian Markov random field by discretizing the continuous domain of the Gaussian process. The primary motivation of this approach is the computational efficiency of the GMRF compared to GP. A sequential prediction algorithm to evaluate a random field, modelled as a Gaussian Markov random field, is proposed by [64]. More details about using the Gaussian Process model is presented in the next chapter. The Gaussian Process model will be used throughout this work to model the signal strength of the Wi-Fi access points inside indoor environments.

1.1.3 Collaborative Approaches to Positioning and Navigation

Standalone localization and positioning systems suffer from several drawbacks. The estimated solution can drift with time when using inertial measurements units (IMU) only. Sometimes the available observations are insufficient for the estimator to work. With fingerprinting techniques, there is a possibility of finding matches in different locations, which gives rise to localization

ambiguities. Collaborative approaches are used to enhance the positioning accuracy and to help to resolve any ambiguities in the estimates. Collaborative techniques have been studied in different contexts, such as in emergency situations for first responders and firefighters and intelligent transportation solutions [65].

A collaborative positioning solution is used for first responders and soldiers [66], using ultra-wideband (UWB) transceivers to measure the range between two mobile nodes and to exchange the position estimates and covariances between participating nodes [67]. The system is modelled according to the standard strap-down mechanization equations [68] and uses 15-state Extended Kalman Filter (EKF) to estimate the position, velocity, orientation, and sensor biases. The selection of this navigation algorithm is justified by the intended short operation period of this system, which is less than ten minutes. The range measurement is included in the EKF update step. The uncertainty of the position estimate of the collaborating nodes is projected to the connecting vector and added to the measurement uncertainty [69] to form the uncertainty of the range measurement equation. However, this formulation ignores the effect of state covariance arising from the collaboration between different nodes [70]. This issue was addressed by using a partially decentralized estimation architecture [71]. In this approach, the estimation problem is divided between two physically separated computational platforms; one is local to the mobile node, and the other is central and shared among all nodes. The local node uses IMU data for dead-reckoning, and update the node position and its variance, using a Kalman filter. The central estimator uses the locally-calculated positions along with the inter-node ranges to correct the positions of the participating nodes. In contrast to the local node estimator, the problem of constraining the position of two nodes with inter-node range is treated as a Bayesian estimation problem [72].

A collaborative approach is used to address the relative positioning problem of vehicles inside urban canyons [73], where the GNSS pseudo-ranges are affected by multipath reflections, causing positioning errors up to tens of meters. The collaboration in this system is done by exploiting the spatial correlation properties of the GNSS pseudo-range measurements between two nearby vehicles [74]. The raw pseudo-range observations are shared among the collaborating vehicles, along with the speed and direction of each vehicle. The positions of the collaborating vehicles are estimated using a single Kalman filter. Additionally, the uncorrelated pseudo-ranges are detected and removed from the measurement update of the relative positioning vector. In this approach, there is no explicit measurement of the ranges between the collaborative nodes. The relative positioning is estimated using a Kalman filter, given the velocities of the vehicles, and further refined by using the spatially correlated pseudo-ranges.

Collaborative localization for wireless sensor networks (WSNs) is a well-established research area, and many of the algorithms and paradigms developed for WSNs localization and positioning can be adapted to smartphone applications [75]–[77]. The spatial and temporal aspects of collaboration between different nodes are defined [78], by showing the structure of the equivalent Fisher information matrix (EFIM) [79], which is used to evaluate the fundamental limits of collaborative positioning algorithms, or the Cramér-Rao Bound (CRB). The CRB is closely related to the geometric dilution-of-precision (GDOP), which is typically used to evaluate the quality of positioning algorithms [75]. The proposed network positioning algorithm depends on a distributed algorithm [76]. The authors examined different non-Bayesian and Bayesian cooperative localization approaches. For non-Bayesian approaches, the authors tested the least-squares (LS) and maximum-likelihood (ML) estimators. The Bayesian approaches depend on minimum mean-square errors (MMSE) or maximum a posteriori (MAP) estimators.

The node localization problem can also be solved as a robust free network adjustment (NA) problem [77]. The network adjustment is performed using iterative re-weighted least squares (IRLS) method, using a robust weighting function.

It is important to note that the early work on cooperative localization in wireless-sensor nodes was mainly focused on stationary nodes [75], and the problem of locating mobile nodes was not the main concern of the related research, in contrast to the scope of the currently proposed research. Nonetheless, the inclusion of these resources is vital as they cover a significant body of knowledge, especially the theoretical aspects of distributed estimation.

Different options are available for inter-node measurements such as received signal strength (RSS), time-of-arrival (TOA), time difference of arrival (TDOA), and angle of arrival (AOA). The time-of-arrival (TOA) is used as a measurement connecting different nodes [80]. The purpose of this system is to estimate the unknown positions of the collaborating nodes and to estimate the unknown channel conditions. This problem can be solved jointly by estimating the position and the channel conditions; or, it can be solved separately by estimating the channel conditions first, then estimating the positions given the channel conditions. In that work, the position and channel conditions are considered independent, to simplify the prediction step. The posterior of the mobile nodes positions and the channel conditions are estimated using a sample importance resampling (SIR) particle filter [81]. To estimate the posterior of the position of a specific node, the node should know the positions of all the mobile nodes and the position of all the anchor nodes, along with the inter-node measurements and the measurements between the nodes and the anchor nodes. This gives rise to two different Bayesian estimation problems; the joint position and channel condition estimation, and the separate position and channel estimation. The channel condition is also treated [82], where additional channel state is investigated, in

which the line-of-sight signal is present, but it may not be distinguishable from the non-line-of-sight one.

Table 1–3 Summary of Collaboration Algorithms for Indoors Navigation

Reference	Node State /Observable	Inter-Node Measurement	Framework
[67]	INS Mechanization	UWB Range	Decentralized – EKF/ZUPT
[71]	INS Mechanization + PDR	UWB Range	Centralized – Bayesian Estimation
[80]	Statistical Model	RF TOA	Decentralized – Particle Filter
[83]	Wi-Fi RSS	Acoustic TOA	Centralized – Graph Rotation + Translation
[84]	PDR	Acoustic Signal – Approximate Location	Decentralized – EKF
[85]	PDR	Acoustic – Proximity	Decentralized – Kalman Filter
[86]	PDR/Wi-Fi RSS	Wi-Fi Direct RSS – Proximity	Horus [33]/Particle Filter

Inter-node ranging system can be achieved using acoustic signals [87]. The system is implemented on cell phones and achieved centimetre accuracy. The distance between two nodes is calculated using the elapsed time between two time-of-arrivals (ETOA), which designate the time of the round trip of an audio signal emitted by the speaker of one of the collaborating nodes, and receiver by the other one. A similar method proposed by [83] utilizes acoustic inter-node ranging and Wi-Fi RSS values of each node to enhance the localization accuracy of all nodes. The method mainly used graph rotation and translation across the Wi-Fi RSS map to determine the location of the collaborating nodes, constrained by the measured inter-node ranges. Another approach for using acoustic signals utilizes a set of single frequency sine waves and assign each frequency to a certain cell on a grid. The exchange starts by mapping the node location to the

corresponding frequency; then, each node sends a beacon with a frequency corresponding to that cell [84]. This approach provides a coarse relative location since the maximum error in the transmitted node position could reach 18 m, and the range information is not measured. Another approach depends on estimating the node position using PDR algorithm [85]. The state is updated using the position of a nearby collaborating node. The proximity detection is done by sensing a high-frequency acoustic signal sent by the collaborating node. Also, in this approach, there is no explicit ranging information provided by the acoustic signal [33].

Social-Loc is a collaborative framework to exploit social interaction between nodes to improve the existing indoor localization techniques [86]. The framework is set atop the standalone localization techniques, such as pedestrian dead-reckoning or Wi-Fi fingerprinting. Social-Loc takes the set of estimated positions of each node individually as an input. By using the interaction information, or the lack of thereof, the framework can eliminate the impossible positions of the collaborating nodes. The inter-node discovery is implemented using Wi-Fi direct RSS, and the detection event occurs when the RSS of another node is above a certain threshold.

1.1.4 Simultaneous-Localization and Mapping for Smartphones

Simultaneous-localization and mapping (SLAM) is a family of algorithms to concurrently estimate the position of a moving node and the map of the surrounding environment–whether this node is a robot, a vehicle, an object or a pedestrian. SLAM is a very active research topic, especially in the robotics community [88]–[90]. SLAM algorithms have been embraced in the navigation and positioning community with a lot of recent papers focusing on using odometry from a foot-mounted inertial navigation unit (IMU) [91], [92], using Wi-Fi ranging or fingerprinting [92]–[98], using magnetic field [99], [100], using radio signals available in the environment [101], or using the recognizable actions of pedestrians as features [102], [103]. It is

important to note that the standard SLAM techniques, such as those used in robotics applications [88], cannot be used directly in the pedestrian navigation case, especially with Wi-Fi or magnetic field fingerprinting techniques, because the standard SLAM algorithms depend on sensing distinct environmental features, either by using LiDAR data or by extracting visual features from images. The nature of Wi-Fi signal strength or magnetic field intensity is different, and it is modelled as a continuous field that changes smoothly with the position, not as a set of discrete features. Additionally, unlike robotics, the motion of pedestrian is not known a priori and could be challenging to model, predict, or control. This section will introduce the main research directions of SLAM algorithms that are geared towards the problem of indoor positioning and localization of smartphones.

An exception to the theme of this section is the range-only method [104], which is an early approach for using SLAM to solve localization and navigation problems. However, this approach is limited by the accuracy of the range measurements, and it is mentioned in this review for completeness.

Perhaps one of the first papers to propose using SLAM techniques to build a map for Wi-Fi-based positioning is the paper by Ferris et al. [93]. They proposed mapping the high-dimensional Wi-Fi signal strength into a low-dimensional latent space, which corresponds to the device location, using the Gaussian Process Latent Variable Model (GP-LVM) [105]. The idea of using GP in Wi-Fi signal strength modelling was first described in [61]. The motion model for this approach is based on a Voronoi graph representation [106], where the environment is represented as a graph, with edges representing environment features–hallways, staircases, or elevators–and vertices connecting between different edges. The same idea is extended by adding polygonal regions to the graph [61], which represent large open areas, such as rooms and halls. The Wi-Fi

SLAM using GP-LVM does not depend on any other sensor; it uses only the Wi-Fi signal strength to estimate the position.

FootSLAM is a Bayesian estimation SLAM framework for pedestrians [91]. The FootSLAM approach depends on the odometry data obtained from a foot mounted IMU. The authors also propose PlaceSLAM algorithm, which adds landmarks proximity sensing to the odometry-only based SLAM. The proximity sensing could depend on an RFID attached to well-known landmarks, or on a camera to detect different visual features of the environment. Finally, they propose FeetSLAM, which adds a cooperative mapping, in the sense of adding multiple datasets to the SLAM problem. The Wi-Fi SLAM algorithm extends the FootSLAM [94] using Wi-Fi received signal strength (RSS) in addition to the odometer data derived from foot-mounted IMU. In Wi-Fi SLAM, the RSS values are modelled by a distance-dependent signal propagation model.

A smartphone-based SLAM algorithm, smartSLAM, is proposed in [107]. The system moves between different operational modes depending on the sensor data available to the mobile node to reduce the computational burden of the localization algorithm. The modes of the algorithms are PDR only; EKF, if the fingerprinting maps are available; EKF-SLAM, uses PDR data along with the Wi-Fi fingerprinting measurements sequence when the map is not available; PF-SLAM, uses particle filter, PDR, fingerprinting data, and magnetic field anomalies.

Wi-Fi fingerprinting-based SLAM is augmented by exploiting the spatial and temporal coherence in sequences of Wi-Fi RSS measurements to detect the loop-closure of a mobile node [96]. This approach is based on the GraphSLAM algorithm [108], intended mainly for offline SLAM.

Magnetic field anomalies are shown to be able to provide global self-localization capabilities [20]. The local anomalies of the ambient magnetic field are used as input to the SLAM algorithm [99], [109], in addition to the odometry measurements. The three orthogonal components of the magnetic vector field are modelled by three independent Gaussian processes, and the pose posterior distribution is evaluated using a Rao-Blackwellized particle filter. Using magnetic field anomalies for indoor pedestrian navigation, using particle filter is proposed in [19]. This approach is an extension to the FootSLAM algorithm–it uses odometry information from foot-mounted IMU, and augment it with magnetic field information, in a similar approach to [94]. Opportunistic sensing of RF signals-of-chance to perform indoor SLAM for pedestrians is proposed in [101]. Signals-of-chance includes television radio signals, commercial radio signals, and cellular networks transmission, in addition to Wi-Fi signals. The authors use a particle filter framework for online estimation of the mobile node position, against an occupancy grid map. ActionSLAM is another method for indoor localization, which utilizes foot mounted IMU to keep track with the user path, and body mounted IMUs (wrist mounted IMU and smartphone in the pocket) to recognize different user actions, which are used as landmarks [102]. ActionSLAM only recognizes action at the corresponding landmark location, making the position association problem a trivial one. ActionSLAM uses an action recognition (AR) block to differentiate between different user's actions and a ZUPT-PDR block to estimate the user's step length and direction. The position posterior is estimated using a particle filter. ActionSLAM estimates the position against a two-dimensional map, and it was extended to estimate position against three-dimensional maps [103].

Table 1–4 Summary of SLAM Algorithms for Indoors Navigation

	Dynamic Model	Observation	Framework
RO-SLAM	Probabilistic Model	Range using TOF/Landmark	Non-Linear Least-Squares Optimization
Wi-Fi SLAM	Probabilistic Model	Wi-Fi RSS	GP-LVM
FootSLAM	Probabilistic Model	PDR/Layout	Particle Filter
WiSLAM	Probabilistic Model	PDR/Layout + Wi-Fi RSS	Particle Filter
MF-SLAM	Odometry (Robot)	Magnetic Field	GP + Particle Filter
MagSLAM	PDR	Magnetic Field	GP + Particle Filter
ActionSLAM	PDR	Action Type	Particle Filter
SmartSLAM	PDR	Wi-Fi, Magnetic Field or Both	EFK, or Particle Filter
Opportunistic Radio SLAM	PDR	Radio Signals: VHF, commercial radio, GSM, Wi-Fi	Particle Filter

1.2 Motivations and Objectives

Reviewing the current literature reveals the significance and the extent of the research done in this field, which is evident by the diversity of solutions and approaches to the indoor positioning and mapping problem. Nevertheless, there are many opportunities to further the current state of the art for indoor positioning and mapping.

One gap of the current literature is the study of the effect of inter-node collaboration on positioning performance. The importance of studying collaboration in indoor navigation is expected to increase in the near future, especially with the introduction of the 5G cellular technology, which has a potential for localization in addition to the support of device-to-device communication [1], [110]–[116].

The main objective of this thesis is to propose and evaluate a new collaborative and multimodal framework for indoor positioning using smartphones to exploit spatial and temporal collaboration between different nodes. The term *multimodal* refers to different navigation aiding

modes, characterized by the different types of mapped features, such as metric and topological floorplans, Wi-Fi access-point location map, Wi-Fi radio map, or magnetic field intensity map. In the context of this work, however, the focus will be mostly on Wi-Fi signal strength maps (radio maps). Furthermore, the term *collaborative* refers to the interaction between different node, which is defined along two dimensions: spatial and temporal. The spatial collaboration takes place when two or more nodes are in physical proximity. In this case, the range or proximity information between different nodes can provide additional constraints to the positioning algorithm, which helps in enhancing the precision of the position estimate for each of the collaborating nodes. The collaboration information can also be used to initialize the position of different nodes. The temporal collaboration occurs through the multimodal maps. This work introduces an efficient map representation to facilitate sharing these maps among the population of navigating nodes or to a centralized node or server.

It is crucial at this point to provide a precise definition of the term *framework*. Essentially, the term framework refers to two things: the statistical model describing the different environmental features and the associated algorithms–which are used to build, update, and optimize this model; and to the inter-node collaborative state estimation algorithm–which describes how to use the inter-node measurements to enhance the positioning accuracy of the collaborative nodes. Given the previous definition of the framework, the proposed research will investigate a single *configuration* of that framework; which defines the map representation, the mapped features, and the associated collaborative estimation algorithms.

1.3 Thesis Outline

Figure 1–4 shows a visual overview of the thesis. The next chapter starts by describing the Gaussian process models in more details; then, it examines the different standalone positioning algorithms based on the GP models. Chapter 3 discusses the new sparse Gaussian process model and evaluates its effect on positioning performance.

In Chapter 4, the UWB-based relative ranging system is introduced, and a study of the achievable positioning performance using this system is discussed as well. A new particle filter family is introduced in Chapter 5, along with the mathematical derivation of the filtering equations and the performance analysis of the different filters. Chapter 6 and 7 present the experimental results of the entire framework and the conclusions.

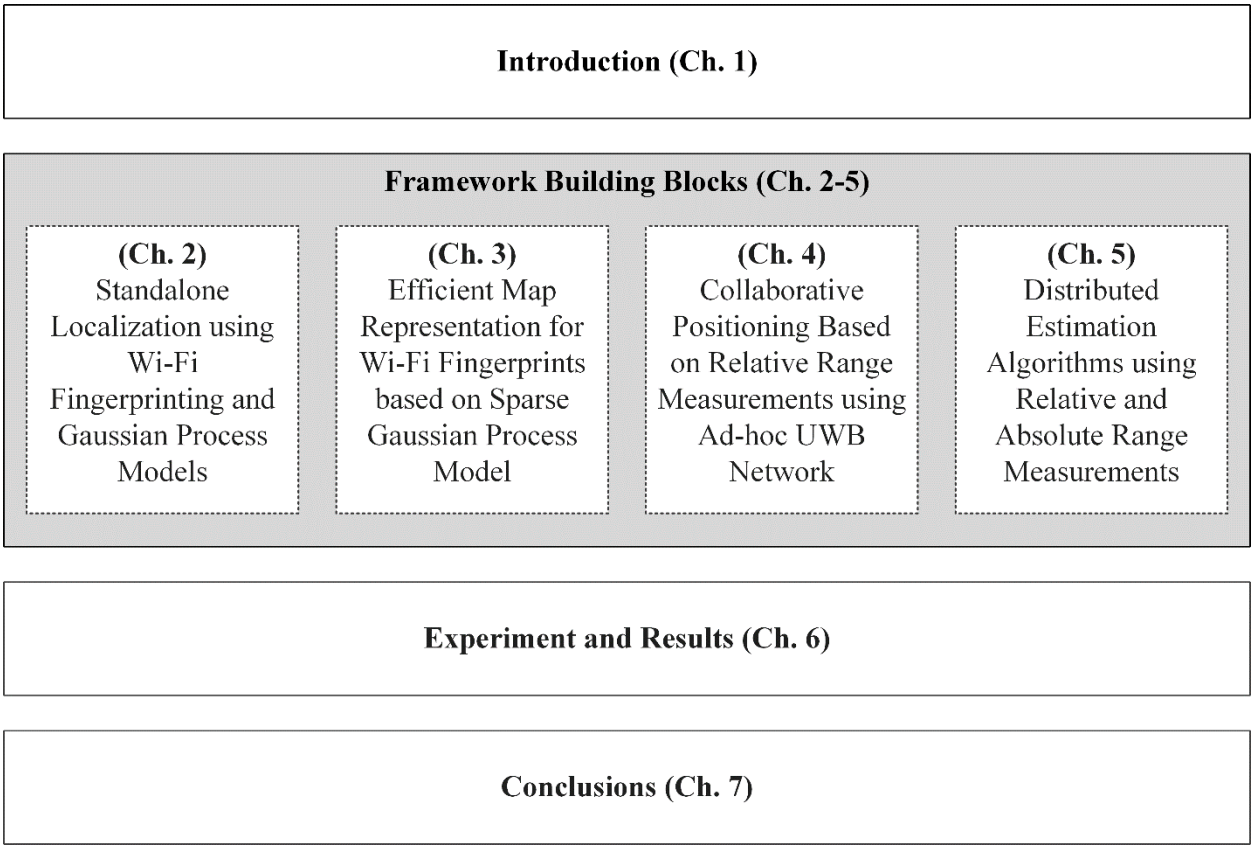


Figure 1–4 Thesis Overview

Chapter Two: **Gaussian Processes for Wi-Fi Fingerprinting**

This chapter starts with a background section, providing an overview of the Gaussian Process (GP) models. Gaussian Process is a non-parametric representation of stochastic processes, and it is the method of choice to represent the Wi-Fi signal strength maps in this work. The next section introduces the Gaussian Process concepts, focusing on GP regression. The chapter proceeds to provide a comparison between different covariance kernel functions, along with a brief discussion about different optimization algorithms for the model hyperparameters.

The problem of position estimation using Wi-Fi fingerprinting is addressed in the following section; specifically, the different filtering techniques used for position estimates using the Wi-Fi signal strength using Gaussian Process model are discussed and compared. The section starts with an overview of the different kinematic models of the state. Then, the measurement update equations and the different filtering approaches are discussed and evaluated.

2.1 Background: Gaussian Process Models Overview

2.1.1 Gaussian Process Regression

Gaussian Process (GP) models are Bayesian tools for regression and classification [117]. GP models are non-parametric [63], i.e. the posterior distribution of the variable of interest is conditioned on the training data. In contrast, in the parametric models, the output is a function of the model parameters; for example, a Gaussian model is parametrized by the mean and variance of the distribution. In this work, the training data for the GP model is composed of tuples, which comprises the observed target value and its corresponding position.

For modelling Wi-Fi signal strength, the GP determines the likelihood of the signal strength conditioned by the position and training data, as shown in Equation (2.1):

$$p\left(y^*_{\mathrm{AP}_i} | x^*, \mathbf{D}_{\mathrm{AP}_i} \right) \tag{2.1}$$

where $y^*_{\mathrm{AP}_i}$ is the estimate of Wi-Fi received signal strength indicator (RSSI) value from access point i at position x^*, and $\mathbf{D}_{\mathrm{AP}_i}$ is the training dataset for the i-th access point. The training data $\mathbf{D}_{\mathrm{AP}_i}$ can be expressed by Equation (2.2):

$$\mathbf{D}_{\mathrm{AP}_i} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \tag{2.2}$$

where x_n is the horizontal position of the training data, $x_n \in \mathbb{R}^2$, y_n is the RSSI value of access point i measured at a point x_n, $y_n \in \mathbb{R}$, N is the total number of training points, and $n \in \{1, \dots, N\}$.

GP models the RSSI value according to Equation (2.3):

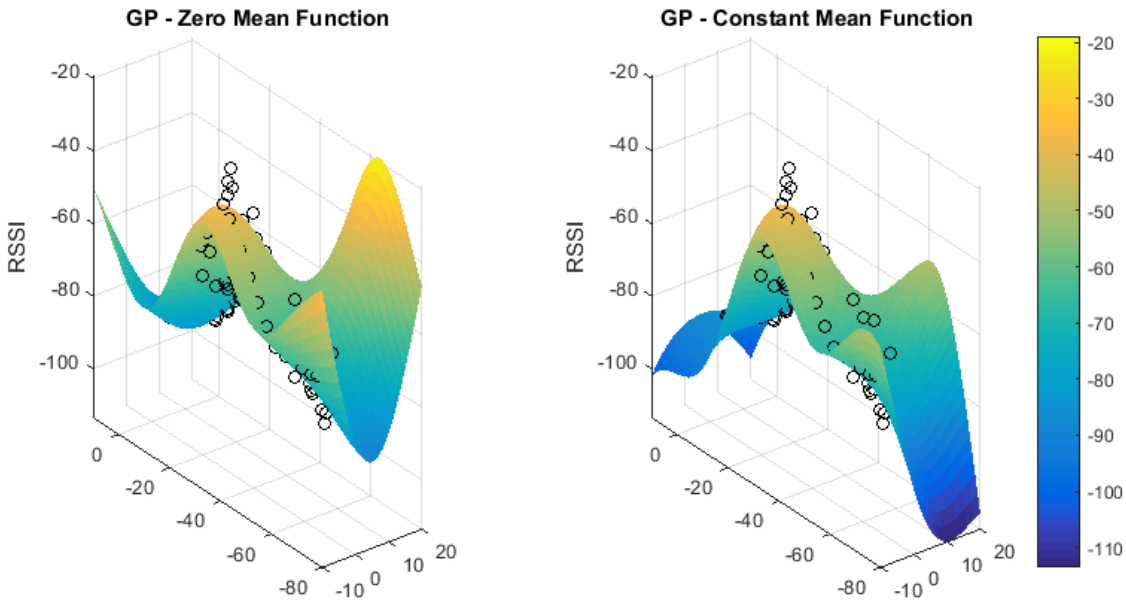
$$y = f(x) + \varepsilon \tag{2.3}$$

where y is the RSSI value, x is the position, ε is a zero-mean Gaussian noise, and $f(x)$ is an unknown, latent function. This formulation does not assume a linear relation between x and y, nor it assumes a Gaussian distribution of y. However, the conditional distribution of y given x, $p(y|x)$, is Gaussian, such that:

$$f(x) \sim \mathcal{N}\big(\mathrm{m}(x), \mathrm{k}(x, x')\big) \tag{2.4}$$

where $\mathrm{m}(x)$ is the process mean function, and $\mathrm{k}(x, x')$ is the covariance function. In other words, Equation (2.4) defines a Gaussian distribution over the function $f(x)$, with mean $\mathrm{m}(x)$ and variance $\mathrm{k}(x, x')$. Any realization of this function may or may not have a mean of zero; however, for any x, the mean of sufficiently large $f(x)$ will converge to $\mathrm{m}(x)$ [118].

Figure 2–1 shows the effect of the GP prior mean function on the posterior distribution. The black circles indicate the locations the values of the training data used to build the RSSI map for a single AP, and the surface indicates the value of the mean of the posterior. Two different mean



(a) (b)

Figure 2–1 Effect of GP Mean Function of the Estimated RSSI Value; (a) zero mean function, (b) constant mean function.

functions are considered: a zero-mean function, and a constant mean function. The zero-mean function will drive the estimated RSSI to zero in regions with no training data available, as shown in Figure 2–1 (a). However, the variance of the estimated RSSI value in these regions will be large enough to account for the lack of training data. However, this behaviour of the mean function does not represent what is expected from the RSSI signal. It is expected that the RSSI signal will decrease as the user moves away from the AP location. One solution is to use a mean function that decreases with increasing the distance between the user and the AP location if the locations of the APs are known [119]. A simple alternative for the mean function would be a constant, set to a low value, resulting in a more realistic estimation of the mean for regions with no training data, as shown in Figure 2–1 (b).

The covariance function or kernel function, $k(x_p, x_q)$, defines the relation between different function values $f(x_p)$ and $f(x_q)$. It indicates how much effect of the training data will have on

the estimated values at new positions. One of the commonly used kernels is called the Radial Basis Function (RBF), or the Squared Exponential Kernel, defined in Equation (2.5):

$$k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\left|x_p - x_q\right|^2\right) \tag{2.5}$$

To account for any noise in the training data, an additional parameter can be added to the kernel function as shown in Equation (2.6):

$$\text{cov}(y_p, y_q) = k(x_p, x_q) + \sigma_n^2 \delta_{pq} \tag{2.6}$$

where δ_{pq} is a Kronecker delta.

The parameter σ_f, l, and σ_n are the GP hyperparameters. The term 'hyperparameters' is used to reassert that these are parameters to a non-parametric model [63]. The objective of learning in the Gaussian Process context is to find the appropriate values of these hyperparameters, given the available training data.

Equation 2.6 can be extended to all N training points, as shown in Equation (2.7):

$$\text{cov}(\mathbf{Y}) = K + \sigma_n^2 I \tag{2.7}$$

where K is an $N \times N$ matrix, where the value of each element is defined as follows, $K[p, q] = k(x_p, x_q)$. Equation 2.1 can be rewritten in terms of the mean and variance functions:

$$p\left(f(x^*)|x^*, \mathbf{X}, \mathbf{Y}_{\text{AP}_i}\right) = \mathcal{N}\left(f(x^*); \mu_{x^*}, \sigma_{x^*}^2\right) \tag{2.8}$$

where AP_i refers to a specific access point, $\mathbf{X}$ is the position of all training data, and $\mathbf{Y}_{\text{AP}_i}$ is the values of training data for a specific access point. The mean and the variance functions are defined by Equations (2.9) and (2.10):

$$\mu_{x^*} = (\text{k}^*)^T (K + \sigma_n^2 I)^{-1} \mathbf{Y}_{\text{AP}_i} \tag{2.9}$$

$$\sigma_{x^*}^2 = k(x^*, x^*) - (\text{k}^*)^T (K + \sigma_n^2 I)^{-1} \text{k}^* \tag{2.10}$$

where k* is the covariance between x^* and the N training points, **X**.

The kernel function is described by a set of hyperparameters, θ, defined as:

$$\theta = \langle \sigma_n, l, \sigma_f \rangle \tag{2.11}$$

The strength of the GP model is that it provides a value estimate–i.e. mean function–along with an estimate of the accuracy of the value–i.e. the variance function. It is worth noting that although the likelihood calculated by Equation 2.8 follows a Gaussian distribution, the mean and variance are functions of the training data; they are not merely constants as in regular Gaussian models.

GP models have been used extensively for indoor localization [61], [120], [121]. Additionally, it can be used with simultaneous localization and mapping algorithms to model the Wi-Fi RSSI values [93]. Moreover, GP could be used to model other environmental features, such as the magnetic field intensity values [100], [109].

2.1.2 The Gaussian Process Covariance Functions

The covariance function describes the similarity between different points in the input space, or how the value of the target variable at a certain point can affect the value of the target function at another point. Different functions can be used to quantify this similarity; some of these functions are described in this section.

The characteristic feature of these covariance functions is that they are stationary; in other words, they are a function of the distance between different points in the input space. The following covariance functions depend on the Euclidian distance between points, r. Where r is defined as:

$$r = \|x - x'\|_2 \tag{2.12}$$

Different examples of the covariance functions are shown in Equations (2.13)–(2.18):

1. Squared Exponential Covariance Functions:

$$k(r) = \exp\left(-\frac{r^2}{2l^2}\right) \tag{2.13}$$

2. Exponential Covariance Functions:

$$k(x_p, x_q) = \exp\left(-\frac{r}{l}\right) \tag{2.14}$$

3. Matérn Class of Covariance Functions:

$$k(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{r}{l}\right)^{\nu} K_{\nu}\left(\sqrt{2\nu}\frac{r}{l}\right) \tag{2.15}$$

where Γ is gamma function and K_{ν} is a modified Bessel function.

a. Matérn 3/2 Kernel

$$k(r) = \left(1 + \sqrt{3}\frac{r}{l}\right)\exp\left(-\sqrt{3}\frac{r}{l}\right) \tag{2.16}$$

b. Matérn 5/2 Kernel

$$k(r) = \left(1 + \sqrt{5}\frac{r}{l} + \frac{5}{3}\frac{r^2}{l^2}\right)\exp\left(-\sqrt{5}\frac{r}{l}\right) \tag{2.17}$$

4. Rational Quadratic Covariance Functions:

$$k(r) = \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha} \tag{2.18}$$

Figure 2–2 shows the estimated Wi-Fi RSSI map for one access point using different covariance functions. Figure 2–2 (a) shows the mean of the RSSI values conditioned on all the training data for this specific access point. Figure 2–2 (b) shows the standard deviation of the estimated RSSI value. The first panel to the left in Figure 2–2 shows the floorplan of the mapped area, with the training positions shown as square points.

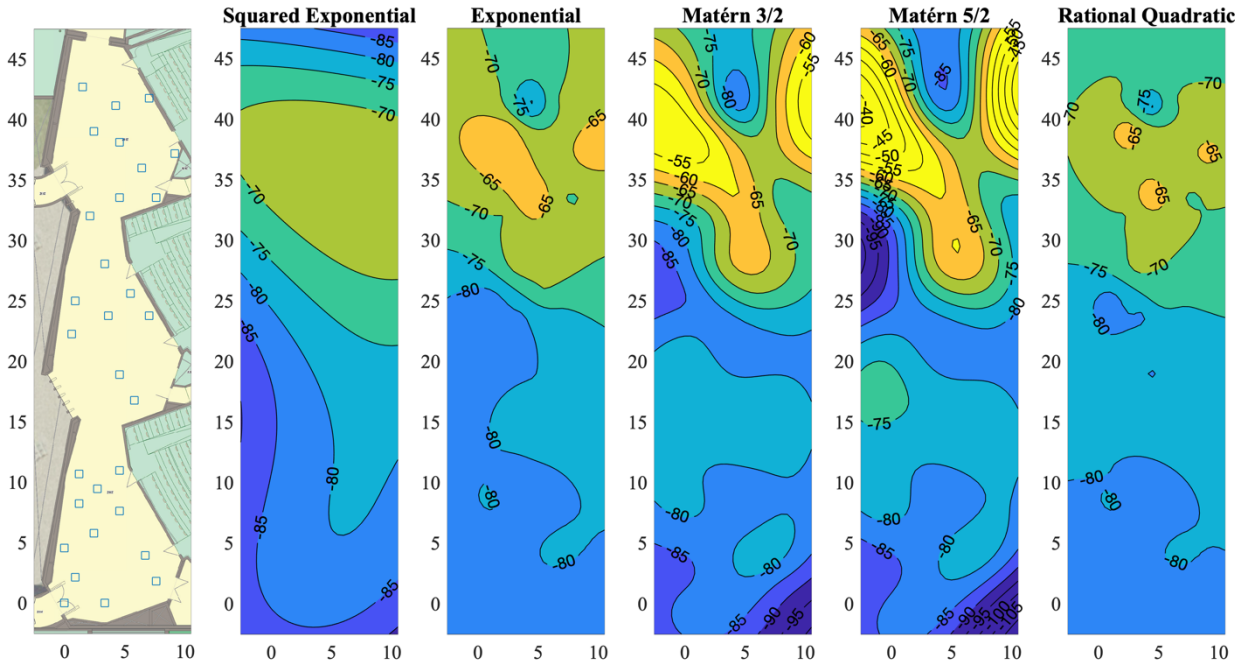
The purpose of the covariance function is to specify the relation between the training set and the target value, or the expected value, at any new position. It is clear that the covariance function

selection affects the behaviour of the expected value, and therefore, the map itself. The squared exponential kernel is the one usually used in related publications, e.g. [93], although without much justification. The rest of this section will examine different kernel functions selection and will attempt to justify the use of the squared exponential function for the rest of this work.

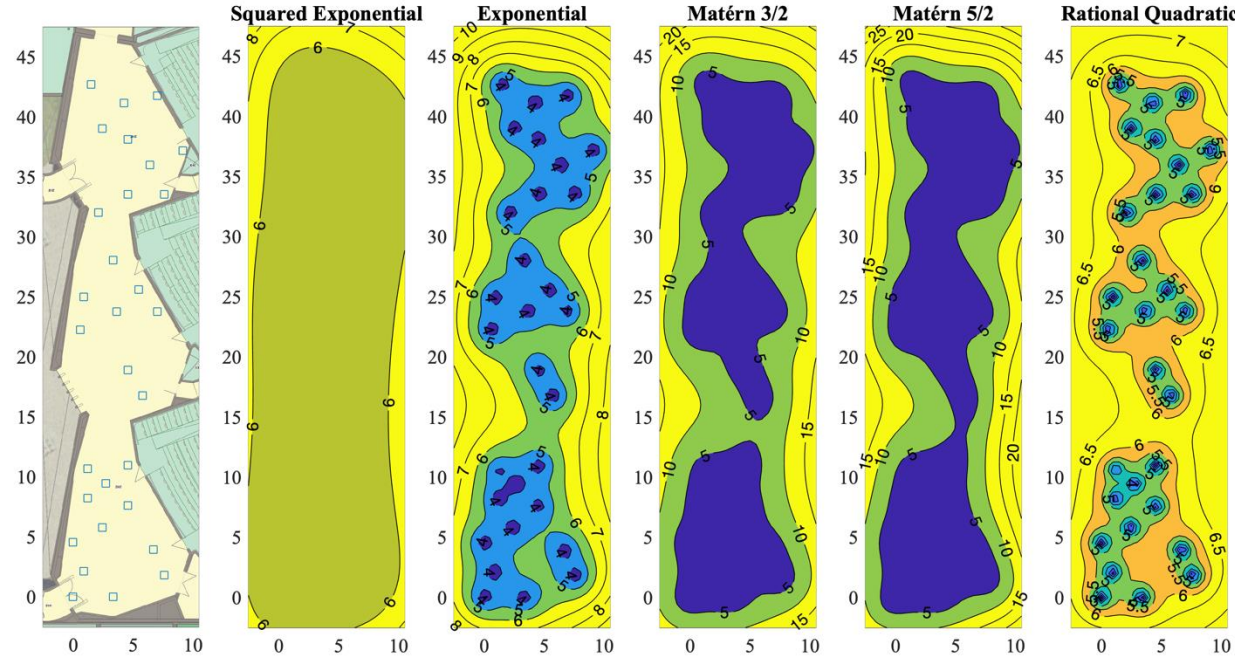
By observing the estimated values in Figure 2–2 (a), it is clear that the squared exponential function closely resembles the expected behaviour of the radio signal strength. The difference between the different maps can be explained by noting the behaviour of the trained kernel function in Figure 2–3. The Rational quadratic kernel and the Exponential kernel are nearly constant over a large distance, that is attributed to the large characteristic length-scale [63], resulting from training the model. In contrast, the value of Matérn kernels drops more quickly.

Another method to evaluate the trained model, and the kernel selection, is by using the K-fold cross-validation method [50]. In this method, the training data is divided into K parts of nearly equal sizes. One of the parts is left out of the training data, and the model is trained using the rest of the data. Then, the model is evaluated using the left part. This process is repeated for K times so that the model is evaluated using each of the K parts. The cross-validation error of the model is the average of the errors from each of the K iterations.

Table 2–1 shows the estimated standard deviation of the Gaussian process trained using the full training set for each kernel function. Table 2–2 shows the RMS error from the K-fold cross-validation of one of the access points RSSI maps. In this case, the K-folds are taken from all the available data. Table 2–3 shows the RMSE for one access point map, trained using the full training data and evaluated using the full training data.



(a)



(b)

Figure 2–2 Wi-Fi RSSI map using different covariance functions; (a) mean (dBm), (b) Standard deviation (dBm)

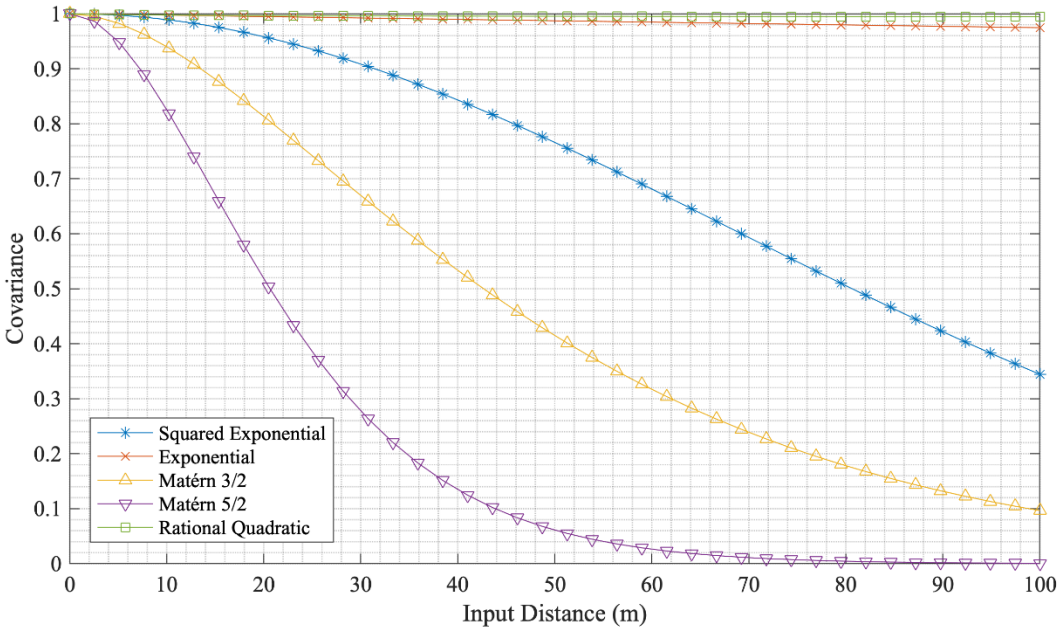


Figure 2–3 Trained covariance functions (kernels)

Table 2–4 shows the average weighted error across all the trained maps. In this case, the K-fold are taking with respect to the training positions. The weighted RMSE is evaluated using Equation (2.19).

$$wrmse = \sum_{i} \left(\frac{1}{\sigma_i} \sqrt{(y_i - \mu_i)^T (y_i - \mu_i)} \right) \tag{2.19}$$

where $wrmse$ is the weighted root-mean-squared error, i refers to the i-th AP map, y_i is the RSSI values vector at the test locations, μ_i is the estimated mean of the RSSI value, and σ_i is the estimated standard deviation of the RSSI value measurements.

Across all the error measures used to evaluate the different covariance functions, the error of the squared exponential kernel is larger than the other kernel. This behaviour can be attributed to the more significant standard deviation of the squared exponential kernel, as shown in Table 2–1. This experiment shows that the exponential, Matérn and rational quadratic kernels tend to be overfitted to the training data, resulting in lower RMSE in the different evaluation scenarios.

Given the previous discussion, the squared exponential kernel appears to be more suitable to the Wi-Fi RSSI training data. It estimates a higher process noise and a moderate length-scale. As a result, the squared exponential kernel is used throughout this work.

Table 2–1 Estimated of the standard deviation of the Gaussian process (dBm)

Covariance Function	STD
Squared Exponential	5.07
Exponential	2.57
Matérn 3/2	2.57
Matérn 5/2	2.56
Rational Quadratic	2.56

Table 2–2 RMSE from a sample map, using K-fold cross-validation (dBm)

Covariance Function	RMSE
Squared Exponential	4.10
Exponential	3.06
Matérn 3/2	3.60
Matérn 5/2	4.16
Rational Quadratic	3.04

Table 2–3 RMSE from a sample AP map, using all the training data (dBm)

Covariance Function	RMSE
Squared Exponential	4.88
Exponential	2.22
Matérn 3/2	2.23
Matérn 5/2	2.22
Rational Quadratic	2.21

Table 2–4 Weighted-RMSE from all APs maps, using K-fold cross-validation

Covariance Function	Weighted RMSE
Squared Exponential	1.36
Exponential	1.05
Matérn 3/2	1.27
Matérn 5/2	1.35
Rational Quadratic	1.04

2.1.3 Hyperparameters Optimization Algorithms

When the training data is available, the value of the hyperparameters can be determined by minimizing the negative-log likelihood function, given by:

$$\mathcal{L} = -\log p(\mathbf{Y}|\mathbf{X}, \theta) \tag{2.20}$$

$$\mathcal{L} = \frac{1}{2} y^T (K + \sigma_n^2 I)^{-1} y + \frac{1}{2} \log|K + \sigma_n^2 I| + \frac{n}{2} \log 2\pi \tag{2.21}$$

This is a non-linear optimization problem, which could be solved using any nonlinear optimization algorithm, including the conjugate gradient descent algorithm, or any quasi-Newton methods such as the symmetric-rank-one (SR1) or the limited-memory BFGS (LBFGS) [122]. To do that, the gradient of the cost function, $\mathcal{L}$, with respect to the hyperparameters, θ, should be calculated, as shown in Equation (2.22):

$$\frac{\partial \mathcal{L}}{\partial \theta_j} = \frac{1}{2} \text{trace} \left((K^{-1} - (K^{-1}y)(K^{-1}y)^T) \frac{\partial K}{\partial \theta_j} \right) \tag{2.22}$$

Using a Radial Basis Function (RBF), expressed in Equation (2.5), with hyperparameters from Equation (2.11), the derivatives of kernel function with respect to each of the hyperparameters, θ_j, are given by Equations (2.23), (2.24), and (2.25):

$$\frac{\partial K_{[p,q]}}{\partial \sigma_f} = 2\sigma_f \exp\left(-\frac{1}{2}\left(\frac{\left\|\mathrm{x}_p - x_q\right\|_2}{l}\right)^2\right) \tag{2.23}$$

$$\frac{\partial K_{[p,q]}}{\partial l} = \sigma_f^2 \exp\left(-\frac{1}{2}\left(\frac{\left\|x_p - x_q\right\|_2}{l}\right)^2\right) \times \left(\frac{\left\|x_p - x_q\right\|_2^2}{l^3}\right) \tag{2.24}$$

$$\frac{\partial K_{[p,q]}}{\partial \sigma_n} = 2\sigma_n \delta_{pq} \tag{2.25}$$

The hyperparameters and the training data fully define the Gaussian Process model.

2.2 Gaussian Process Based Filter Architectures

This section investigates different filter architectures for indoor positioning using Wi-Fi received signal strength indicator (RSSI), modelled as Gaussian Process. Different filter architectures are implemented to assess the complexity of using GP for indoor positioning, and their performance and computational complexity are evaluated. The filter architecture can be divided into two components: the kinematic model, and the measurement update stage. Two different kinematic models are presented using 2-D and 3-D state vectors. The measurement update stage is implemented using Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF). Additionally, to reduce the computational burden of evaluating the GP, the effect of using a reduced version of the GP training data set will be investigated.

Three evaluation criteria are used to compare the different architectures: positioning accuracy, computation complexity, and convergence behaviour. First, the positioning accuracy of the proposed filters is evaluated against a reference trajectory. The computation complexity of different filters is evaluated using the simulation run-time. Finally, the convergence behaviour of the different filters is discussed.

Section 2.2.1 introduces two different kinematic models that could be used to update the state of the system in the absence of external measurements updates. Section 2.2.2 introduces the different filter architectures evaluated in this chapter. The simulation environment and the results are presented in Section 2.2.3.

2.2.1 Kinematic Modeling

This section reviews two different kinematic models for the mobile nodes, which will be used in the time-update section of the localization filters. Two different approaches are presented: full mechanization equations, and the pedestrian dead-reckoning approach.

2.2.1.1 Full Mechanization Equations

The standard mechanization equations are used for the kinematic modelling of inertial navigation systems (INS) [123]–[125]. In this formulation, the position, velocity, and orientation of the mobile node are represented by Equations (2.26), (2.27), and (2.28):

$$\dot{r}^l = \begin{bmatrix} \dot{\phi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} 0 & (R_M + h)^{-1} & 0 \\ \left((R_N + h)\cos\phi\right)^{-1} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_e \\ v_n \\ v_u \end{bmatrix} = D^{-1} V^l \tag{2.26}$$

$$\dot{V}^l = R_b^l f^b - \left(2\Omega_{ie}^l + \Omega_{el}^l\right) V^l + g^l \tag{2.27}$$

$$\dot{R}_b^l = R_b^l \left(\Omega_{ib}^b - \Omega_{il}^b\right) \tag{2.28}$$

where ϕ, λ, h is the latitude, longitude, and height of the node; V^l is the velocity vector in the local-level frame; v_e, v_n, v_u is the node velocity in east, north, and up directions, respectively; R_b^l is the rotation matrix from the body frame to the local-level frame; R_m, R_n are the two principal radii of curvature of the earth; g^l is the normal gravity vector; f^b is the specific force measured in the body frame; and Ω_{ib}^b is the skew-symmetric angular rate measured in the body frame.

The error state in the three-dimensional state vector case can be represented by perturbing the system propagation matrix defined by Equations (2.26), (2.27), and (2.28), around the most recent state estimate. The error state is given by the following equations [124]:

$$\begin{bmatrix} \delta\dot{\phi} \\ \delta\dot{\lambda} \\ \delta\dot{h} \end{bmatrix} = \begin{bmatrix} 0 & (R_M + h)^{-1} & 0 \\ \left((R_N + h)\cos\phi\right)^{-1} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta v_e \\ \delta v_n \\ \delta v_u \end{bmatrix} \tag{2.29}$$

$$\begin{bmatrix} \delta\dot{v}_e \\ \delta\dot{v}_n \\ \delta\dot{v}_u \end{bmatrix} = \begin{bmatrix} 0 & f_u & -f_n \\ -f_u & 0 & f_e \\ f_n & -f_e & 0 \end{bmatrix} \begin{bmatrix} \delta p \\ \delta r \\ \delta y \end{bmatrix} \tag{2.30}$$

$$\begin{bmatrix} \delta\dot{p} \\ \delta\dot{r} \\ \delta\dot{y} \end{bmatrix} = \begin{bmatrix} 0 & (R_M + h)^{-1} & 0 \\ -(R_N + h)^{-1} & 0 & 0 \\ -\tan\phi/(R_N + h) & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta v_e \\ \delta v_n \\ \delta v_u \end{bmatrix} \tag{2.31}$$

The state errors are evaluated using measurement update or using motion constraints [126].

2.2.1.2 Pedestrian Dead-Reckoning

Pedestrian dead-reckoning (PDR) is a technique to calculate the position in two-dimensional spaces by detecting and accumulating the steps of the mobile node. Generally, the PDR algorithm is divided into two parts: step detection and position update. The step detection uses the vertical acceleration component and utilizes the Equation (2.32) to calculate the step length [127]:

$$S_{SL} = \sqrt[4]{d_2} \times K \tag{2.32}$$

where K is a step scaling constant, and d_2 is the difference between the peak and the valley of the upward acceleration, as shown in Figure 2–4, where the y-axis is the magnitude of the vertical acceleration, and the x-axis is the sample number in the processing buffer.

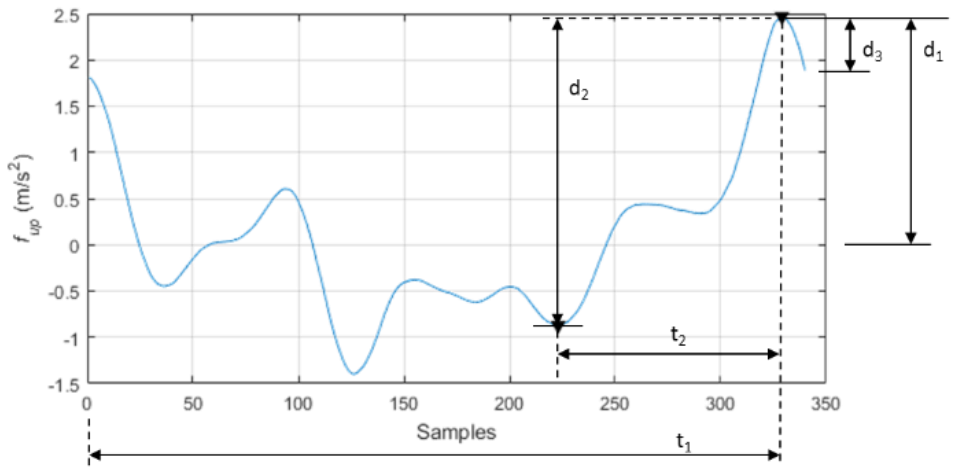


Figure 2–4 Step Detection

A step is detected when all the criteria defined in Table 2–5 are met. These criteria are set to ensure proper spacing between consecutive peaks and to ensure the selection of the prominent peak in the current search window.

Table 2–5 Step Detection Criteria

A step is detected if all the following conditions are met	
d1	> Min. Peak Value
d2	> Min. Peak Difference
d3	> Min. Peak Difference
t1	> Min. Peak Distance
t2	< Min. Peak Distance

The full state of each node is the horizontal position (latitude and longitude) and the orientation of the mobile node (rotation quaternion), as shown in Equation (2.33):

$$x_k = [\phi_k \quad \lambda_k \quad {}^{l}_{b}q_k] \tag{2.33}$$

where x is the state vector, ϕ and λ is the node position, ${}^{l}_{b}q$ is the rotation quaternion, and k is the current time step.

The state transition equation in the local-level frame is given by the Equation (2.34):

$$\begin{bmatrix} \phi_{k+1} \\ \lambda_{k+1} \\ {}^{l}_{b}q_{k+1} \end{bmatrix} = \begin{bmatrix} \phi_k + \dfrac{SL_k \times \cos(A)}{(R_m + h)} \\ \lambda_k + \dfrac{SL_k \times \sin(A)}{(R_n + h) \times \cos(\phi)} \\ {}^{l}_{b}q_k + \dfrac{1}{2}\left(\bar{\omega}_{\times} \times {}^{l}_{b}q_k \times \delta t\right) \end{bmatrix} \tag{2.34}$$

where SL_k is the step length given by equation (2.32), h is the height of the mobile node, which is kept constant, A is the heading angle, δt is the time step, and $\bar{\omega}_{\times}$ is the skew-symmetric matrix of the augmented rotation vector. The error state is modelled as the error in the east and north position and the heading error, Equation (2.35).

$$\delta x_k = [\delta E_k \quad \delta N_k \quad \delta A_k]^T \tag{2.35}$$

Linearizing the state propagation model will result in the error state transition matrix, shown in Equation (2.36).

$$\begin{bmatrix} \delta E_{k+1} \\ \delta N_{k+1} \\ \delta A_{k+1} \end{bmatrix} = \begin{bmatrix} \delta E_k + SL_k \cos(A_k)\, \delta A_k \\ \delta N_k - SL_k \sin(A_k)\, \delta A_k \\ \delta A_k \end{bmatrix} \tag{2.36}$$

2.2.2 Positioning Filter Architectures

This section describes the different filtering approaches used for indoor localization using RSSI modelled using Gaussian Process models. It gives a brief overview of the filtering algorithm and shows the detailed measurement update equations for Wi-Fi RSSI updates.

2.2.2.1 Extended Kalman Filter

Kalman filter is an optimal state estimation algorithm that minimizes the root-mean-square error of linear systems with zero-mean noise. However, if the system is non-linear, there are several suboptimal approximations to the Kalman filter. One of the simplest approximations is the Extended Kalman filter (EKF) [128], [129]. In EKF, the probability density function (PDF) of

the state is propagated using one point, i.e. the mean of the PDF. The general form of the system and the measurements equations are given by Equations (2.37) and (2.38):

$$x_k = f(x_{k-1}, v_k) \tag{2.37}$$

$$y_k = h(x_k, w_k) \tag{2.38}$$

where $v_k \sim \mathcal{N}(0, Q_k)$ is a zero-mean Gaussian process noise with variance Q_k, while $w_k \sim \mathcal{N}(0, R_k)$ is the measurement noise. An overview of the extended Kalman filter algorithm is described by Algorithm 2–1.

ALGORITHM 2–1 EKF TIME UPDATE AND MEASUREMENT UPDATE ALGORITHM

1. The non-linear system equation is linearized around the most recent estimate of the state vector ($\hat{x}_{k-1}^{+}$).

$$F_k = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_{k-1}^{+}} \tag{2.39}$$

$$G_k = \left.\frac{\partial f}{\partial v}\right|_{x=\hat{x}_{k-1}^{+}} \tag{2.40}$$

2. The prior state estimate and the prior covariance matrix are calculated using the non-linear system model, and the linearized system model.

$$\hat{x}_k^{-} = f(\hat{x}_{k-1}) \tag{2.41}$$

$$P_k^{-} = F_k P_{k-1}^{+} F_k^T + G_k Q_{k-1} G_k^T \tag{2.42}$$

3. When a new measurement is available, the measurement equation is linearized around the prior state estimate–the best available estimate of the state at this point.

$$H_k = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_k^{-}} \tag{2.43}$$

$$M_k = \left.\frac{\partial h}{\partial w}\right|_{x=\hat{x}_k^{-}} \tag{2.44}$$

4. The Kalman gain is calculated, and the updated state estimate along with the state covariance are calculated according to the following equations.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \tag{2.45}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k\left(y_k - h(\hat{x}_k^-)\right) \tag{2.46}$$

$$P_k^+ = (I - K_k H_k) P_k^- \tag{2.47}$$

5. Repeat from step 1.

The measurement equation for the GP based Wi-Fi estimation is equal to the mean of the likelihood function expression, given by Equation 2.9. In this case:

$$h(x_k) = \mu_{x_k} \tag{2.48}$$

The linearization of the measurement equation around x_k is given by the Equations (2.49), (2.50), and (2.51):

$$H_k = \frac{\partial}{\partial x}\left(\mathrm{k}^T (K + \sigma_n^2 I)^{-1}\mathbf{Y}\right)\Big|_{x=\hat{x}_k^-} \tag{2.49}$$

$$H_k = \left(\frac{\partial}{\partial x}\mathrm{k}\Big|_{x=\hat{x}_k^-}\right)^T \times (K + \sigma_n^2 I)^{-1}\mathbf{Y} \tag{2.50}$$

$$\frac{\partial}{\partial x}\mathrm{k}\Big|_{x=\hat{x}_k^-} = \frac{1}{l^2}\begin{bmatrix} k_{x,x_0}(x_0 - x) \\ \vdots \\ k_{x,x_m}(x_m - x) \\ \vdots \\ k_{x,x_M}(x_M - x) \end{bmatrix}_{x=\hat{x}_k^-} \tag{2.51}$$

Note that in the previous equations, K and $\mathbf{Y}$ do not depend on x. They are both function only of the training data.

2.2.2.2 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) differs from EKF in the state PDF propagation method. In UKF the state propagation occurs using multiple points, called sigma points, instead of only one in EKF case. The estimated state is computed as a weighted sum of these sigma points [130], [131].

In order to reduce the run-time of the UKF, the measurement update is implemented using the UKF algorithm, while the state update is done using the linearized version, similar to the EKF algorithm. The implemented UKF algorithm for performing the measurement update against the GP map is summarized in Algorithm 2–2.

ALGORITHM 2–2 UKF MEASUREMENT UPDATE ALGORITHM

1. The state update is performed similarly to the EKF; Algorithm 2–1, steps 1-2.
2. When a new measurement is available, calculate sigma points:

$$\chi_0|_{k+1} = \hat{x}_{k+1}^-$$

$$\chi_i|_{k+1} = \hat{x}_{k+1}^- + \left(\sqrt{(N+\lambda)\hat{P}_{k+1}}\right)_i \qquad i = 1, \dots, N \tag{2.52}$$

$$\chi_i|_{k+1} = \hat{x}_{k+1}^- - \left(\sqrt{(N+\lambda)\hat{P}_{k+1}}\right)_{i-N} \qquad i = N+1, \dots, 2N$$

$$\gamma = \sqrt{N+\lambda} \tag{2.53}$$

$$\lambda = \alpha^2(N+\kappa) - N \tag{2.54}$$

where N is the number of states, and κ, α are parameters that control the spread of the sigma points around the mean of the state estimate.

3. Evaluate the expected Wi-Fi RSSI value and its variance using the GP model, using Equation (2.48), for each sigma point i:

$$Z_{k+1}^{[i]} = h\left(\chi_{k+1}^{[i]}\right) \tag{2.55}$$

4. Evaluate the weights of each point:

$$w_m^0 = \frac{\lambda}{L+\lambda} \tag{2.56}$$

$$w_c^0 = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \tag{2.57}$$

$$w_m^i = w_c^i = \frac{1}{2(L+\lambda)} \qquad i = 1, \dots, 2N \tag{2.58}$$

where β is a parameter related to the distribution of the variable x, for a Gaussian distribution, the optimal value of β is 2 [130].

5. Calculate predicted measurement:

$$\hat{z}_{k+1} = \sum_{i=0}^{2N} w_m^{[i]} Z_{k+1}^{[i]} \tag{2.59}$$

6. Calculate measurement covariance and the measurement and state cross-correlation:

$$P_{k+1}^{zz} = \left(\sum_{i=0}^{2N} w_c^{[i]} \left(Z_{k+1}^{[i]} - \hat{z}_{k+1}\right)\left(Z_{k+1}^{[i]} - \hat{z}_{k+1}\right)^T\right) + R_{k+1} \tag{2.60}$$

$$P_{k+1}^{xz} = \sum_{i=0}^{2n} w_c^{[i]} \left(\chi_{k+1}^{[i]}\right)\left(Z_{k+1}^{[i]} - \hat{z}_{k+1}\right)^T \tag{2.61}$$

7. Evaluate Kalman gain and update the state estimate along with the state covariance:

$$K = P_{k+1}^{xz}(P_{k+1}^{zz})^{-1} \tag{2.62}$$

$$\hat{x}_{k+1}^{+} = \hat{x}_{k+1}^{-} + K(z_{k+1} - \hat{z}_{k+1}) \tag{2.63}$$

$$P_{k+1} = \hat{P}_{k+1} - K P_{k+1}^{zz} K^T \tag{2.64}$$

8. Repeat from step 1.

2.2.2.3 Particle Filter

The particle filter (PF) is a non-linear state estimator; but, unlike the EKF and UKF, it uses the full non-linear system and measurement model, not just a first-order approximation such as the EKF, or a higher-order approximation such as the UKF. The PF approximates the posterior distribution of the state by a weighted set of particles [129], [132].

Initialization of the particle filter is done by sampling M equally weighted particles from the known prior distribution of the particles $p(x_0)$. The weights of the particles are updated based on the external measurement likelihood at each particle. Algorithm 2–3 gives an overview of a simple version of the particle filter algorithm.

ALGORITHM 2–3 PARTICLE FILTER ALGORITHM

1. At $k = 0$, initialize M particles and their weights. The initial particles are sampled from the prior distribution $p(x_0)$. At initialization, the weights of all particles are equal.

$$x_{1,i}^{+} \sim p(x_0) \tag{2.65}$$

$$w_{1,i} = \frac{1}{M} \tag{2.66}$$

2. For all next iterations, $k > 1$, sample $v_{k,i}$ for M times from the system noise distribution and perform the time update using the non-linear system model:

$$v_{k,i} \sim \mathcal{N}(0, Q_k) \tag{2.67}$$

$$x_{k,i}^{-} = f\left(x_{k-1,i}^{+}, v_{k,i}\right) \tag{2.68}$$

3. When a new measurement is available, update the weights of the particles, according to the likelihood of observing the new measurements given each particle location:

$$w_{k,i} = \frac{\left(w_{k-1,i} p\left(y_k | x_{k,i}\right)\right)}{\left(\sum_{i=1}^{M} w_{k-1,i} p\left(y_k | x_{k,i}\right)\right)} \tag{2.69}$$

4. The weighted particles approximate the posterior PDF, according to:

$$\hat{p}(x_k|y_k) = \sum_{i=1}^{M} w_{k,i}\delta(x_k, x_{k,i}) \tag{2.70}$$

5. The estimated state could be approximated as the mean of the particles:

$$\hat{x}_k \approx \sum_{i=1}^{M} w_{k,i}x_k \tag{2.71}$$

6. Calculate the effective number of samples, M_{eff}:

$$M_{eff} = \frac{1}{\sum_{i=1}^{M}(w_{k,i})^2} \tag{2.72}$$

7. If the effective number of samples is less than a certain threshold, M_{th}, resample the particles from the approximate distribution. The resampling is done by taking M samples with replacement from the set of particles $\{x_{k,i}, w_{k,i}\}$ to produce $\{x_{k,i}, 1/M\}$.

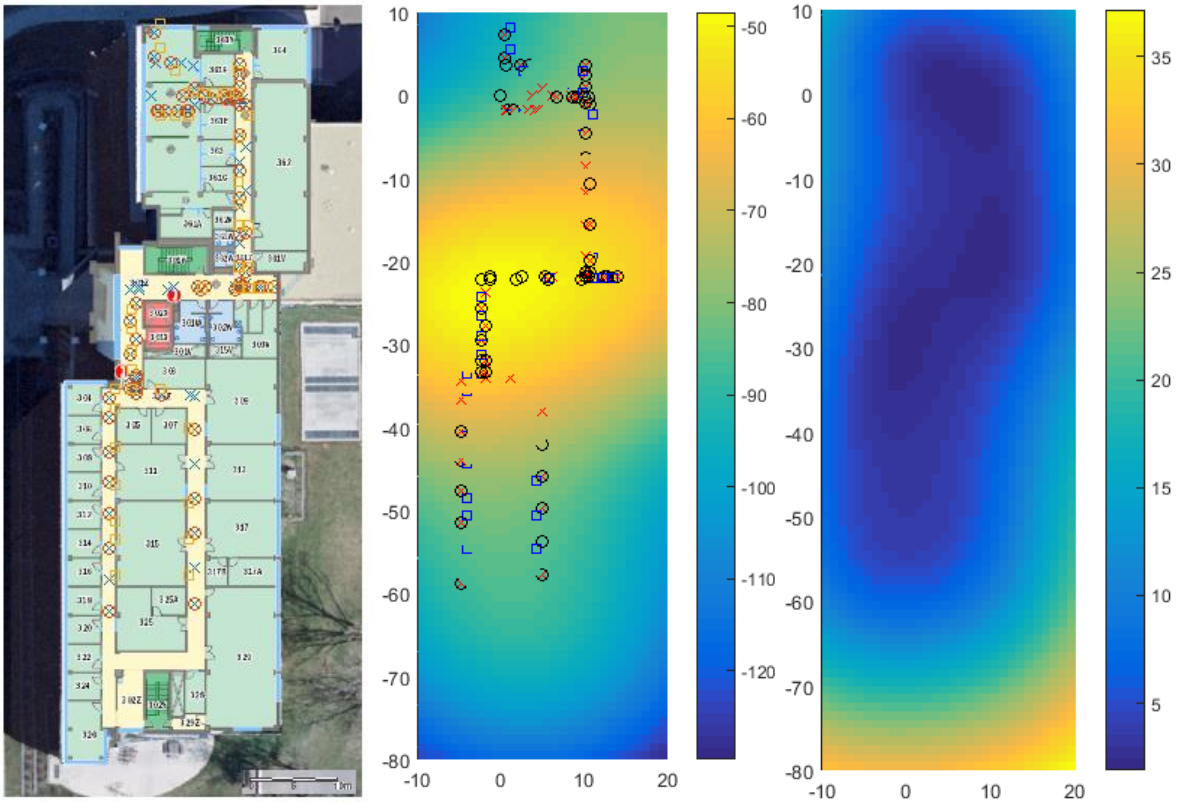
8. Repeat starting from step 2.

The resampling step is performed to overcome the particle depletion problem, where the weight of most of the particles tends to zero, so they do not contribute to the posterior distribution. In this case, the number of effective particles, that describes the state PDF is small. Resampling generates particles according to the estimated posterior distribution.

2.2.3 Experimental Results

The performance of the different filters is evaluated with data collected using a smartphone, in one of the buildings at the University of Calgary. The data is collected using a smartphone held steadily by the user to maintain the relative orientation between the smartphone and the user. The

ground-truth trajectory and the Wi-Fi reference map are calculated and used to evaluate the performance of the proposed filter architectures. The total length of the trajectory is 270 m, and the duration of the trajectory is 7 minutes. The testing area is located on the third floor of the CCIT building at the University of Calgary. The floorplan of the testing area is shown in Figure 2–5 (a). The mean and variance of one instance of the Wi-Fi RSSI map is shown in Figure 2–5 (b) and (c).



(a) (b) (c)

Figure 2–5 Reference Map: (a) floorplan, (b) mean of Wi-Fi RSSI map for one access point, (c) the variance of the Wi-Fi AP RSSI map

The system models are evaluated first, then the performance of the filters is evaluated according to the localization accuracy, the run-time, and convergence characteristics of the filter.

2.2.3.1 Reference Trajectories Generation

The reference trajectory of this experiment is along the centre line of the corridors travelled by the mobile node, as shown in Figure 2–7 (a). It is clear that the trajectory is composed of straight-line segments with sharp rotations between consecutive segments. However, this definition of the reference is missing the time information, needed to compare between the reference and the estimated trajectory. The temporal information of the reference trajectory is generated by performing a piecewise two-dimensional similarity transformation of the PDR trajectory, to align the time-stamped PDR trajectory with the known trajectory; thus, providing the needed time component to each point on the reference trajectory.

Figure 2–6 shows the transformation of three sample segments to establish the time correspondence between the PDR trajectories and the reference trajectories. The time inside a

Figure 2–6 Reference trajectory generation

single segment is synchronized in two stages: using the steps detection time; then, in between steps, the time is linearly interpolated.

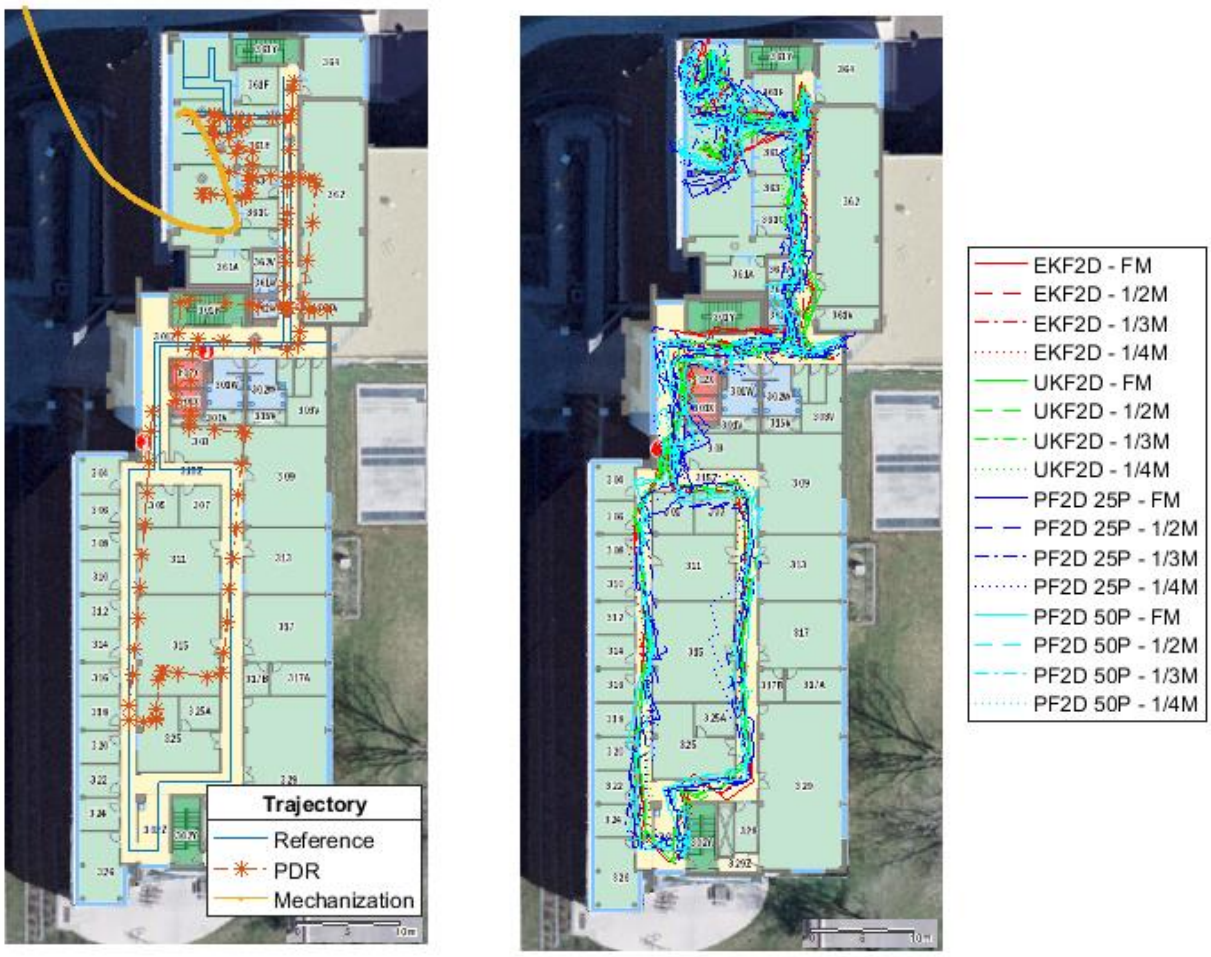
2.2.3.2 System Models Performance

Figure 2–7 (a) shows the reference trajectory, the PDR trajectory, and the mechanization trajectory. It is clear that the error from the mechanization equation is increasing with large rate. Additional constraints could be used to enhance the performance of the mechanization equation, including using PDR system models [126]. However, for the rest of this section, only the PDR system model will be used to evaluate the performance of the different filter architectures.

2.2.3.3 Localization Filter Performance

Figure 2–7 (b) shows the trajectories generated using different filters, with different settings for the Wi-Fi map. This section compares the performance of the EKF, UKF, and particle filters, using a different number of training points in the map; this will be referred to as a full map, a half-map, a third-map, and a quarter-map. These settings are denoted as FM, 1/2M, 1/3M, and 1/4M, respectively. The map reduction is achieved by dropping a point each two, three or four points from the complete training data set; and since the training data is collected sequentially, dropping points will not result in losing much information from the Wi-Fi map. There are two versions of the particle filter, one uses 50 particles to approximate the posterior distribution, while the other uses 25 particles.

The cumulative distribution function (CDF) of the position error is shown in Figure 2–8. Also, the value of the mean, RMS, and maximum error in position along with the run time of each filter are shown in Figure 2–9 and in Table 2–6. The EKF and UKF can achieve similar positioning errors between 2.4 m to 2.8 m (RMS) while having very similar run times. The two particle filters can achieve positioning accuracy up to 1.5 m, with more processing time.



(a) (b)

Figure 2–7 Trajectories: (a) reference and system model, (b) state estimates

The effect of changing the map size is not significant on the positioning accuracy or on the total run time. The first observation can be explained by the fact that consecutive training points are correlated and dropping a few of them will not significantly affect the accuracy of the Wi-Fi map. Decreasing the map size results in decreasing the processing time; however, this decrease is not significant as expected with the $\mathcal{O}(N^3)$ complexity. This can be explained by the overhead in the algorithm computation and control code, which dominates the execution time.

2.2.3.4 Filter Convergence Characteristics

In all simulations shown earlier the initial state of the different filters was assumed to be known within 1~2 m. Figure 2–10 shows the position error when the initial value is not known. In the EKF and UKF filters, this is achieved by setting the initial position to the map middle and increase the initial state covariance matrix. In the particle filter, this is done by choosing a uniform prior over the map area for the filter states.

The particle filter can converge to its regular error levels within a few seconds of operation; after the first measurement update, the particle filter estimate was near the correct trajectory. On the other hand, EKF and UKF were able to estimate the correct trajectory eventually; however, it took them multiple measurements updates to converge to the correct estimation.

Table 2–6 Position Error and Run-Time

Filter	**Map**	**Position Error (m)**			**Time (s)**	
		MEAN	*RMS*	*MAX*		
PDR	-	5.7	6.5	12.5	-	-
EKF2D	100%	2.1	2.5	8.0	103	100%
EKF2D	50%	2.1	2.6	8.2	101	98%
EKF2D	33%	2.1	2.6	8.9	100	97%
EKF2D	25%	2.3	2.8	8.8	102	99%
UKF2D	100%	2.1	2.4	6.8	133	129%
UKF2D	50%	2.1	2.4	7.4	124	120%
UKF2D	33%	2.2	2.6	7.1	121	117%
UKF2D	25%	2.2	2.6	8.5	120	116%
PF2D 25P	100%	1.6	1.9	5.6	395	382%
PF2D 25P	50%	1.6	1.8	4.8	361	349%
PF2D 25P	33%	1.6	1.9	5.1	351	340%
PF2D 25P	25%	1.5	1.7	4.0	346	335%
PF2D 50P	100%	1.3	1.6	4.2	600	581%
PF2D 50P	50%	1.4	1.6	4.0	538	520%
PF2D 50P	33%	1.3	1.5	3.6	517	501%
PF2D 50P	25%	1.3	1.5	4.4	507	491%

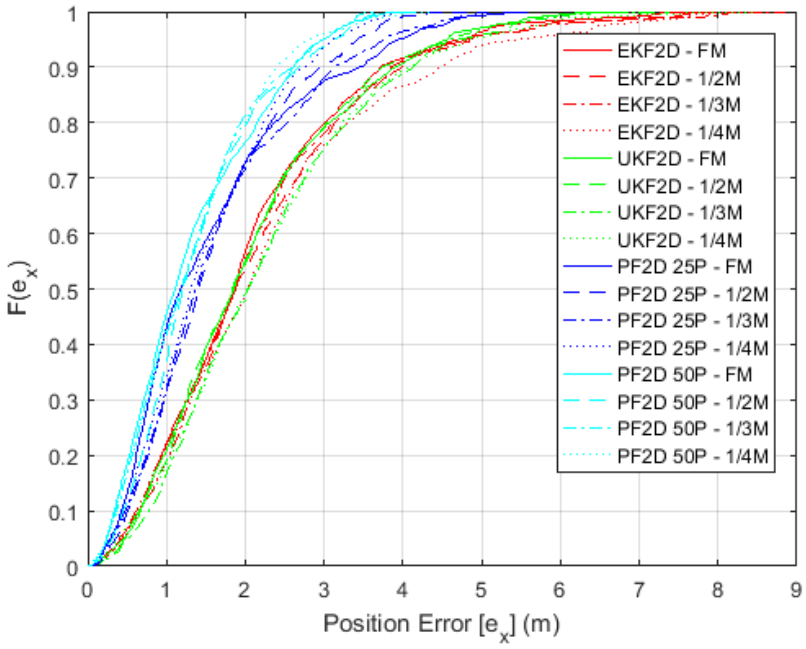


Figure 2–8 Position Error CDF

2.3 Summary

This chapter presented an overview of the Gaussian process models, their use for regression, the different kernel functions that can be used to code the relation between the different points in the input space of the model, and the different optimization approaches that can be used to find the hyperparameters of the Gaussian process model.

The chapter then addressed the problem of positioning using environmental feature maps built using the Gaussian process models. Different kinematic models and measurement update formulations were implemented and evaluated. The positioning performance, the execution time, and the convergence characteristics for each filter were evaluated and compared.

This chapter defines the first component of the proposed framework: the standalone positioning filter. It describes the internal states of each node, the supported class of observables, and the state update algorithm for each node.

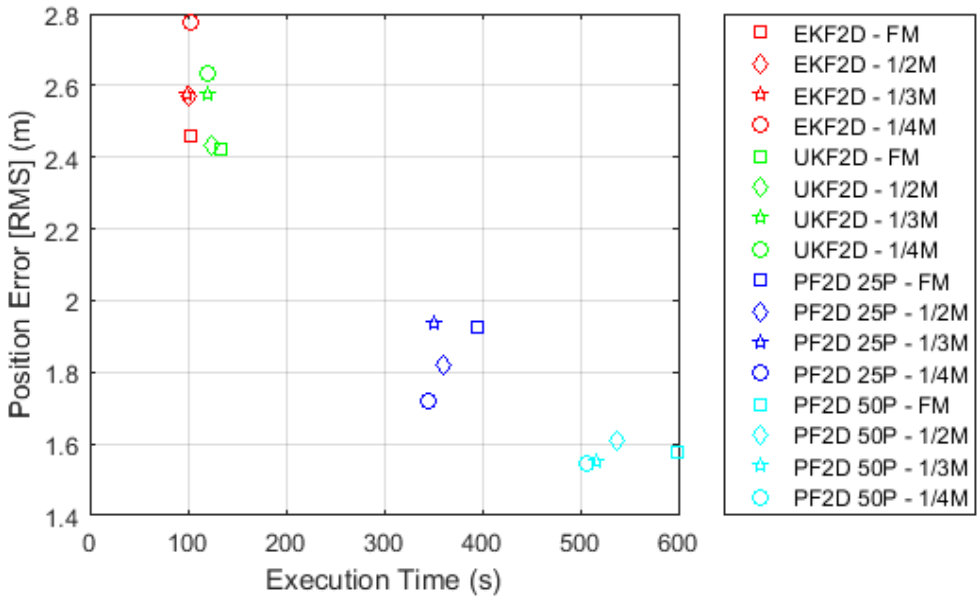


Figure 2–9 Position Error and Run-Time

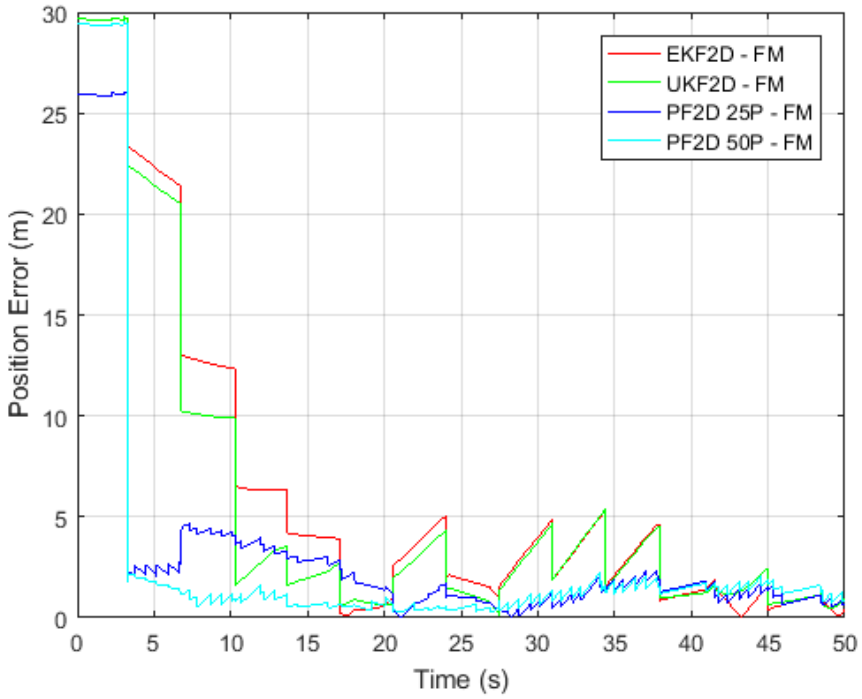


Figure 2–10 Position Error Convergence

Chapter Three: **Efficient Wi-Fi Fingerprints Map**

This chapter starts by providing a brief overview of different Gaussian process sparsification algorithms. The following section introduces a new sparsification algorithm for Gaussian process models, named the parametric grid sparse Gaussian process model. In this algorithm, the target values of a set of pseudo-input data, defined over a subset of a parametric grid, are selected and optimized to represent the whole Gaussian process model using fewer training points. The details of the algorithm, along with the performance evaluation, in terms of positioning performance and computational complexity reduction, are presented and discussed.

3.1 Background: Sparse Gaussian Process Overview

The computation complexity of using a GP model to estimate the target value at a new input point is dominated by the calculation of the covariance matrix inverse, K^{-1}, which is an $N \times N$ matrix, where N is the size of the training dataset. The computation complexity of the matrix inverse operation is in the order of $\mathcal{O}(N^3)$, while the most efficient algorithm can achieve $\mathcal{O}(N^{2.3727})$ complexity [133]. The complexity reduction of GP-based regression has motivated the effort to find an approximation to the full Gaussian process model [63]. Another equally important factor when using GP for indoor positioning application is the storage limitation and the bandwidth constraints imposed by using a mobile device. As a result, one of the main objectives of the proposed GP sparsification algorithm is to decrease the size of the transmitted dataset from a remote server to the mobile devices.

The different GP approximations can be divided into two main categories: covariance matrix approximation and the likelihood function approximation [63], [134]–[137]. The likelihood function approximation is further divided into the fully independent training conditional (FITC) approximation and the deterministic training conditional (DTC) approximation. In the DTC

approach, a representative subset of the training dataset is used; in addition to several algorithms for the insertion and deletion of training points to the representative data subset. On the other hand, the FITC approach uses a set of virtual inputs, and optimize the location of these inputs along with the model hyperparameters and the target value at the location of the virtual inputs [134].

The sparse GP approach is used for target tracking [138], where a fixed grid based recursive GP algorithm is introduced. In this algorithm, the posterior distribution is evaluated locally over a fixed grid; then, a global map is built by fusing the local maps in a centralized data center. The main difference between this algorithm and the one presented in this work is that the grid in the proposed approach is not fixed. Instead, it is parametrized to offer more flexibility in the pseudo-point selection.

3.2 Parametric Grid Sparse Gaussian Process Model

This section proposes and evaluates a new algorithm to increase the computation and storage efficiency and to reduce the bandwidth requirements of the Wi-Fi received signal strength indicator (RSSI) maps based on Gaussian process (GP) models. The GP models are non-parametric models, in which the likelihood of the target variable is conditioned on a set of training data. The Parametric Grid Sparse GP (PGSGP) algorithm objective is to improve the efficiency of using GP maps. The PGSGP reduces the complexity of evaluating the likelihood function, by reducing the number of points in the training dataset, without significant loss of the mapping or positioning accuracy. This reduction is achieved by finding a set of pseudo-inputs arranged over a parametric grid, then optimizing the corresponding target values and the GP model hyperparameters to maximize the likelihood function of observing the actual training dataset.

Section 3.2.1 starts by discussing the details of GP optimization based on pseudo-input. Section 1.2.2 outlines the proposed sparsification algorithm. Finally, section 3.3 presents the simulation environment and discusses the results.

3.2.1 Pseudo-Input Gaussian Process

The objective of the sparsification algorithm is to find a set of pseudo-inputs and their corresponding target values that maximize the joint likelihood function of the RSSI values observed at the location of the training points. In order to accomplish that goal, the likelihood function of each of the training point is expressed in terms of the pseudo-inputs set. Then, the joint likelihood function of the entire training set is evaluated. Finally, the negative log-likelihood function is optimized to obtain the GP hyperparameters along with the pseudo-inputs position and target values. The likelihood function of a single training point, given a set of pseudo-inputs and targets, is given by Equation (3.1) [134].

$$p\left(y\middle|x,\theta,\bar{\mathbf{X}},\bar{\mathbf{f}}\right)=\mathcal{N}\left(y\middle|\mathrm{k}_x^T\mathrm{K}_M^{-1}\bar{\mathbf{f}},K_{xx}-\mathrm{k}_x^T\mathrm{K}_M^{-1}\mathrm{k}_x+\sigma^2\right) \tag{3.1}$$

where $\bar{\mathbf{X}}\in\mathbb{R}^{M\times 2}$ is the pseudo-input set, and $\bar{\mathbf{f}}\in\mathbb{R}^{M\times 1}$ is the pseudo-target set.

$$[\mathrm{k}_x]_m=k(\bar{x}_m,x) \tag{3.2}$$

$$[\mathrm{K}_M]_{mm'}=k(\bar{x}_m,\bar{x}_{m'}) \tag{3.3}$$

where $m=1,\dots,M$, and $k(x,x')$ is the covariance function.

The training inputs are independent, so the likelihood function of the N-points training dataset is the product of the individual points likelihood functions:

$$\begin{aligned}p\left(\mathbf{y}\middle|\mathbf{X},\theta,\bar{\mathbf{X}},\bar{\mathbf{f}}\right)&=\prod_{n=1}^{N}p\left(y_n\middle|x_n,\theta,\bar{\mathbf{X}},\bar{\mathbf{f}}\right)\\&=\mathcal{N}\left(\mathbf{y}\middle|\mathrm{K}_{NM}\mathrm{K}_M^{-1}\bar{\mathbf{f}},\boldsymbol{\Lambda}+\sigma^2\mathbf{I}\right)\end{aligned} \tag{3.4}$$

where $\mathbf{X} \in \mathbb{R}^{N\times 2}$ is the position of the training dataset, and $\mathbf{y} \in \mathbb{R}^{N\times 1}$ is the training target values.

$$[\mathrm{K}_{NM}]_{nm} = k(x_n, \bar{x}_m) \tag{3.5}$$

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda) \tag{3.6}$$

$$[\lambda]_n = K_{nn} - \mathrm{k}_n^T \mathrm{K}_M^{-1} \mathrm{k}_n \tag{3.7}$$

The following equations describe the mean vector and variance matrix:

$$\mathrm{M} = \mathrm{K}_{NM}\mathrm{K}_M^{-1}\bar{\mathbf{f}} \tag{3.8}$$

$$\Sigma = \mathbf{\Lambda} + \sigma^2\mathbf{I} \tag{3.9}$$

The optimization of the GP hyperparameters, pseudo-inputs and pseudo-targets follows the same path as the optimization of the full GP. In this case, the conjugate gradient descent algorithm is used to minimize the negative log-likelihood function, given by Equation (3.10).

$$\mathcal{L} = -\log p\left(\mathbf{y}\middle|\mathbf{X}, \theta, \bar{\mathbf{X}}, \bar{\mathbf{f}}\right) \tag{3.10}$$

$$\mathcal{L} = \frac{1}{2}(\mathbf{y} - \mathrm{M})^T\Sigma^{-1}(\mathbf{y} - \mathrm{M}) + \frac{1}{2}\log|\Sigma| + \frac{N}{2}\log 2\pi \tag{3.11}$$

The gradient of the loss function is calculated with respect to the elements of the GP hyperparameters vector, the pseudo-inputs, and the pseudo-targets.

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\theta_j} = \frac{1}{2}\mathrm{trace}\left(\Sigma^{-1}\left(\frac{\partial}{\partial\theta_j}\Sigma\right)\right) \\ -\alpha^T\left(\frac{\partial}{\partial\theta_j}\mathrm{M}\right) - \frac{1}{2}\mathrm{trace}\left(\alpha\alpha^T\left(\frac{\partial}{\partial\theta_j}\Sigma\right)\right)\end{aligned} \tag{3.12}$$

where, $\alpha = \Sigma^{-1}(\mathbf{y} - \mathrm{M})$, and $\theta_j \in \{\sigma_n, l, \sigma_f, \bar{x}_1, \dots, \bar{x}_M, \bar{f}_1, \dots, \bar{f}_M\}$.

The derivative of the mean and covariance functions with respect to the optimization parameter is given by Equations (3.13) and (3.14).

$$\frac{\partial}{\partial\theta_j}\mathrm{M} = \left(\frac{\partial}{\partial\theta_i}\mathrm{K}_{NM}\right)\mathrm{K}_M^{-1}\bar{\mathbf{f}} + \mathrm{K}_{NM}\mathrm{K}_M^{-1}\left(\frac{\partial}{\partial\theta_i}\mathrm{K}_M\right)\mathrm{K}_M^{-1}\bar{\mathbf{f}} + \mathrm{K}_{NM}\mathrm{K}_M^{-1}\left(\frac{\partial}{\partial\theta_i}\bar{\mathbf{f}}\right) \tag{3.13}$$

$$\left[\frac{\partial}{\partial\theta_j}\Sigma\right]_{nn} = \left(\frac{\partial}{\partial\theta_j}K_{nn}\right) - 2\mathrm{k}_n^{\mathrm{T}}\mathrm{K}_M^{-1}\left(\frac{\partial}{\partial\theta_j}\mathrm{k}_n\right) - \mathrm{trace}\left((\mathrm{K}_M^{-1}\mathrm{k}_n)(\mathrm{K}_M^{-1}\mathrm{k}_n)^T\left(\frac{\partial}{\partial\theta_i}\mathrm{K}_M\right)\right) + 2\sigma\left(\frac{\partial}{\partial\theta_j}\sigma\right) \tag{3.14}$$

The derivatives of the K matrix with respect to the GP hyperparameters are given earlier.

The derivatives of the loss function with respect to the pseudo-inputs and pseudo-targets are given by Equations (3.15) to (3.18):

$$\frac{\partial}{\partial\bar{x}_m}\mathrm{K}_{NM} = \frac{1}{l^2}\begin{bmatrix} 0 & \dots & 0 & k(x_1,\bar{x}_m)\times(x_1-\bar{x}_m) & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & k(x_N,\bar{x}_m)\times(x_N-\bar{x}_m) & 0 & \dots & 0 \end{bmatrix} \tag{3.15}$$

$$\frac{\partial}{\partial\bar{x}_m}\mathrm{K}_M = \frac{1}{l^2}\times \begin{bmatrix} \mathbf{0} & k_{i,m}\times(\bar{x}_i-\bar{x}_m) & \mathbf{0} \\ k_{m,i}\times(\bar{x}_i-\bar{x}_m) & 0 & k_{m,j}\times(\bar{x}_j-\bar{x}_m) \\ \mathbf{0} & k_{m,j}\times(\bar{x}_j-\bar{x}_m) & \mathbf{0} \end{bmatrix} \tag{3.16}$$

$$\frac{\partial}{\partial\bar{x}_m}\mathrm{k}_n = \frac{1}{l^2}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ K_{n,m}(x_n-\bar{x}_m) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.17}$$

$$\begin{cases} \left[\frac{\partial}{\partial\bar{f}_i}\bar{\mathbf{f}}\right]_{j=i} = 1 \\ \left[\frac{\partial}{\partial\bar{f}_i}\bar{\mathbf{f}}\right]_{j\neq i} = 0 \end{cases} \tag{3.18}$$

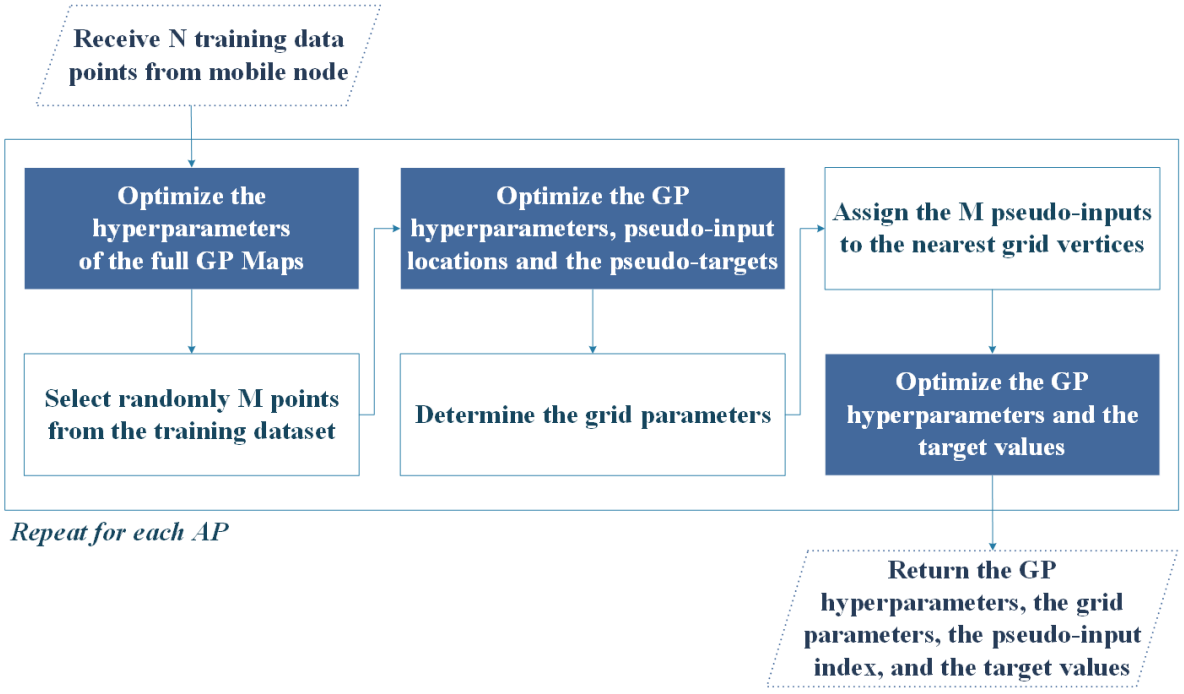


Figure 3–1 PGSGP Algorithm Overview

As will be elaborated in the next section, the optimization of the sparse GP is performed twice. During the first optimization run, all parameters are allowed to change. While in the second run the positions of the pseudo-input are fixed. In the latter case, the derivative with respect to $\bar{x}_m$ is set to zero, and the GP hyperparameters and pseudo-inputs are the only parameters allowed to change.

3.2.2 The Algorithm Overview

Figure 3–1 shows an overview of the proposed sparsification algorithm. The input to the algorithm is a set of N training data points. The objective is to obtain a set of M pseudo-inputs, where $M < N$, along with the GP hyperparameters and the target values at the pseudo-input set. The M pseudo-inputs are constrained to be on the vertices of a parametric grid. Constraining the pseudo-inputs to the vertices of a parametric grid alleviates the need to store or transmit the

locations of the set. Instead, a remote server needs only to send the grid parameters, along with a set of indices tying the location of each of the pseudo-inputs to one of the grid vertices.

ALGORITHM 3–1 PARAMETRIC GRID SPARSE GAUSSIAN PROCESS (PGSGP) ALGORITHM:

1. Obtain N training points.
2. Optimize the negative-log likelihood function, to obtain the initial set of the GP hyperparameters $\theta_{GP} = \langle \sigma_n, l, \sigma_f \rangle$.
3. Initialize the position of M points, by randomly assigning each one of them to the location of one of the training data [134].
4. Optimize the GP hyperparameters, the pseudo-inputs positions and the target values, using (3.11). This step is similar to the FITC algorithm [134].
5. Determine the parameters of the irregular grid.

   For each dimension in the pseudo-inputs:

   1. Arrange the elements of the i-th dimension in a vector.
   2. Sort the i-th dimension elements in ascending order.
   3. Merge the i-th dimension elements with less than 1 m of separation.
   4. Find the coefficients of a second order curve that fit the i^{th} dimension elements.
6. Assign the pseudo-inputs to the nearest grid point.
7. Fix the pseudo-inputs and optimize the GP hyperparameters and the target values.
8. Return the GP hyperparameters, the grid parameters, the pseudo-input indices, and the target values.

Figure 3–2 shows the transformations occurring to the training dataset at several steps of the algorithm. The result of the first optimization run–Algorithm 3–1, Step 2–is illustrated in the

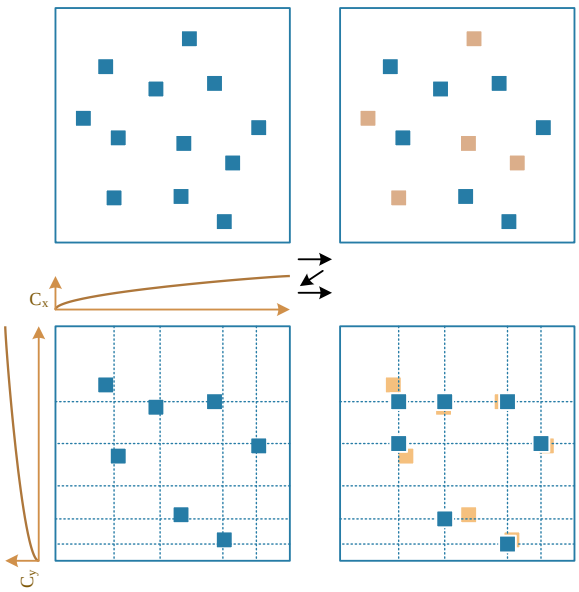


Figure 3–2 Pseudo-Inputs Locations

upper-left quadrant of Figure 3–2, where the locations of the N training points is not changing. Then, M pseudo-points are selected, and their locations are initialized randomly to the locations of M points from the original training dataset–Step 3. The other points are discarded as shown in the upper-right quadrant of Figure 3–2, where the discarded points are marked in orange. After the selection and initialization of the pseudo-inputs, another optimization run is performed–Step 4. As a result, the initial positions might move slightly to maximize the likelihood function. The x- and y- positions of the pseudo-inputs are fitted to two second-order curves, $(\mathbf{C}_x, \mathbf{C}_y)$, as shown in the lower-left quadrant. Finally, as shown in lower-right quadrant, the pseudo-inputs are fixed to the grid vertices, then the final optimization run determines the GP parameters and pseudo-target values–Step 7.

3.3 Experiments and Results

The performance of the algorithm is evaluated using multiple criteria: the reduction in the required storage and accordingly the bandwidth; the loss in mapping accuracy, relative to the reference map; and the effect of the sparsification algorithm on the positioning accuracy.

Table 3–1 summarizes the reduction in size from using the PGSGP algorithm compared with using a subset of the training dataset. The target value is assumed to be stored in a one single-precision variable, while the position in the subset of data case is assumed to be stored in two single-precision variables. On the other hand, for the PGSGP algorithm, the position is stored as an index to the parametric grid vertices, and it uses a single integer variable. However, the grid parameters are stored as well in the PGSGP approach and consume six float variables. The parameters are the coefficients of two second-order curves, approximating the locations of the pseudo-input locations. This table shows clearly the advantage of using the PGSGP algorithm. In the first row, without even reducing the number of points representing the map, approximately 50% reduction in the storage size could be achieved.

Table 3–1 PGSGP Size Reduction

No. of Points	Subset of Data				PGSGP			
	y	X	Total (bytes)	Gain	y	X	Total (bytes)	Gain
90	360	720	1080	-	360	180	546	49%
60	240	480	720	33%	240	120	366	66%
50	200	400	600	44%	200	100	306	72%
40	160	320	480	56%	160	80	246	77%
30	120	240	360	67%	120	60	186	83%
20	80	160	240	78%	80	40	126	88%
10	40	80	120	89%	40	20	66	94%

The mapping error is evaluated by calculating the Root Mean Square Error (RMSE) between the estimated values and the reference values, at the locations of the training dataset. Another measure used is the Kullback-Leibler Divergence (KLD), which defines the distance, or the dissimilarity, between two probability distributions. In this case, the distributions are the one

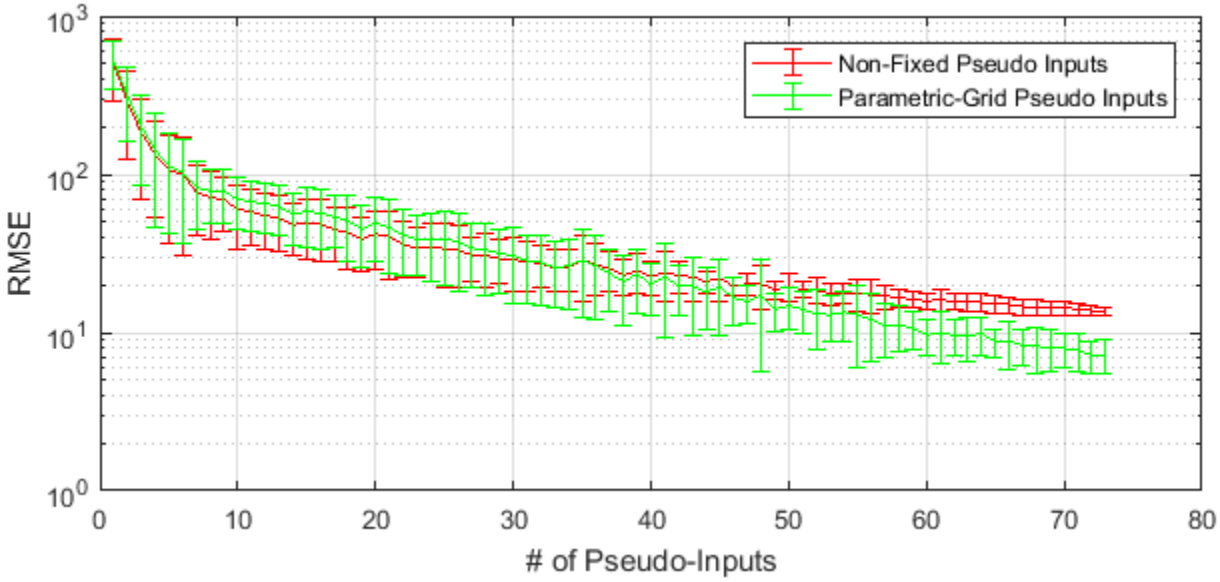


Figure 3–3 Root Mean Square Error

obtained by using the reference full-GP, and the one obtained by using PGSGP. The KLD between two distributions is defined by Equation (3.19).

$$\mathrm{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx \tag{3.19}$$

For a Gaussian distribution, the KLD is given by Equation (3.20) [63].

$$\begin{aligned} \mathrm{KL}(\mathcal{N}_0\|\mathcal{N}_1) &= \frac{1}{2}\log|\Sigma_1\Sigma_0^{-1}| \\ &+ \frac{1}{2}\mathrm{trace}\left(\Sigma_1^{-1}\left((\mathrm{M}_0 - \mathrm{M}_1)(\mathrm{M}_0 - \mathrm{M}_1)^T + \Sigma_0 - \Sigma_1\right)\right) \end{aligned} \tag{3.20}$$

Figure 3–3 and Figure 3–4 show the RMSE and the KLD as a function of the number of pseudo-inputs. These curves are generated using multiple instances of the PGSGP maps, using different subsets of the training points as initialization to the algorithm. The two figures show the mean and the 1-σ bar for the RMSE and the KLD.

The maximum number of iterations for the conjugate gradient descent optimization algorithm is set to 50 iterations. The general trend of the curve shows that as the number of points increases

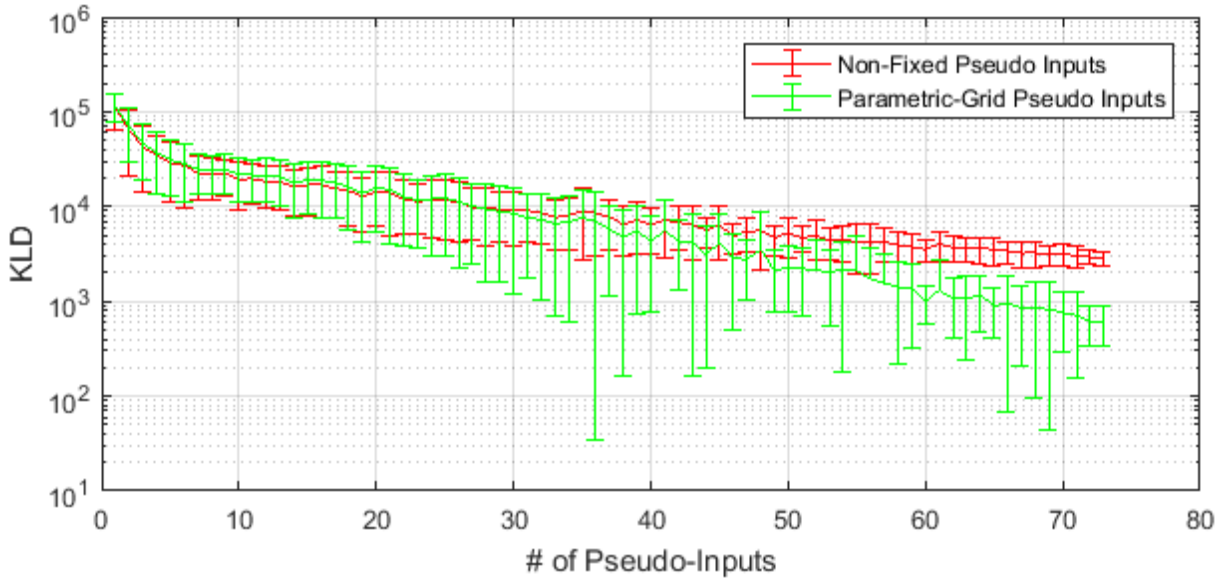


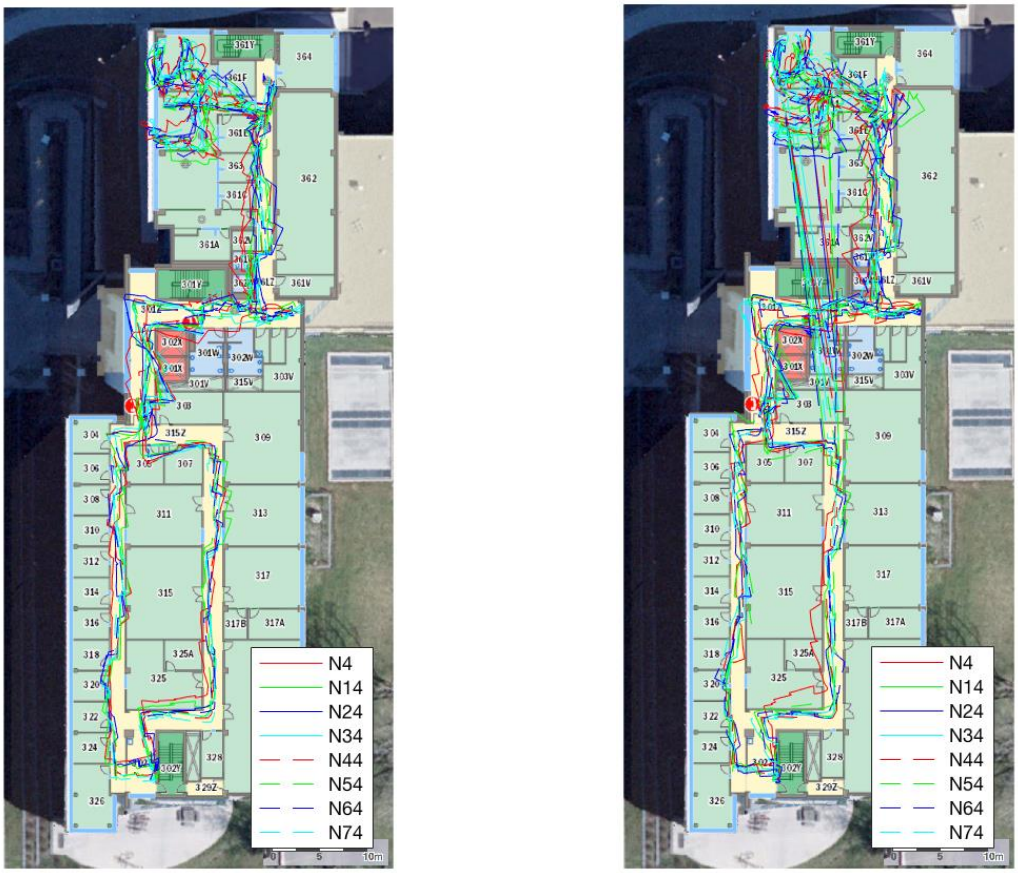
Figure 3–4 Kullback-Leibler Divergence

the RMSE and KLD tends to decrease, which is expected; since using more points means that the approximated GP can capture more of the features of the reference map. The performance of the PGSGP is better than the unfixed inputs when the number of pseudo-inputs is about 55% of the full training dataset. It worth noting that the random assignment of the pseudo-inputs initial position causes the performance of the PGSGP to fluctuate around the mean. This fluctuation becomes more pronounced when the number of points decreases to a small fraction of the original dataset, because the randomly selected initial points may not be representative to the full likelihood function.

The previous discussion focused mainly on different measures to quantify the mapping errors, resulting from applying the proposed sparsification algorithm. However, in mapping applications, the main concern would be the positioning error after applying the sparsification algorithm. Figure 3–5 shows the positioning results using different numbers of points in each access point map. The number indicates the maximum number that could be used for each map; when an access point has less training points than the limit, all available points are used. The

positioning is performed using a particle filter with 50 particles. Figure 3–5 (a) shows the results assuming the initial position is known, basically using a Gaussian prior on the initial particles with a mean equal to the correct starting position and a small variance. Figure 3–5 (b) shows the results with a uniform prior over the entire area of interest; this is done to test the convergence performance of the filter and ensure that the solution will eventually converge to the correct path. Figure 3–6 shows the cumulative distribution function (CDF) of the positioning errors from the Gaussian prior case. Each line in Figure 3–6 is associated with the number of points used is the Wi-Fi RSSI maps. For example, 'N4' means that the maximum number of points used in all maps is 4. Table 3–2 lists the mean, the root-mean-square errors, and the maximum errors for the two priors.

The general trend in the positioning errors shows an increase in the RMSE and the mean error with the reduction in the maximum number of points in each access point map, which is congruent with the expected behaviour. The main point to notice is that 50% of the errors across a wide range of points (4 to 74) is concentrated between 1-2 m. This can be explained by the fact that the area of interest has a dense Wi-Fi access point coverage. As a result, even when dropping most of the training points randomly, there would be enough points–across all available access points–to provide good positioning performance. Additionally, the non-monotonic decrease in the positioning error that could be observed in Table 3–2 is attributed to the random selection policy of the training points.



(a) (b)

Figure 3–5 Positioning Results; (a) Gaussian initialization, (b) uniform initialization

Table 3–2 PGSGP Positioning Error

Max. No. of Points	Initial Particle Distribution					
	Normal			Uniform		
	Mean (m)	*RMS (m)*	*MAX (m)*	*Mean (m)*	*RMS (m)*	*MAX (m)*
4	1.88	2.06	4.09	1.95	3.26	29.48
14	1.49	1.72	4.57	1.84	3.84	38.14
24	1.52	1.73	4.49	1.72	2.96	26.78
34	1.27	1.50	3.62	1.68	3.30	32.27
44	1.32	1.59	4.65	1.58	3.29	32.64
54	1.27	1.48	4.12	1.82	3.72	36.79
64	1.42	1.67	4.66	1.81	3.72	35.81
74	1.27	1.54	4.71	1.64	3.54	35.33

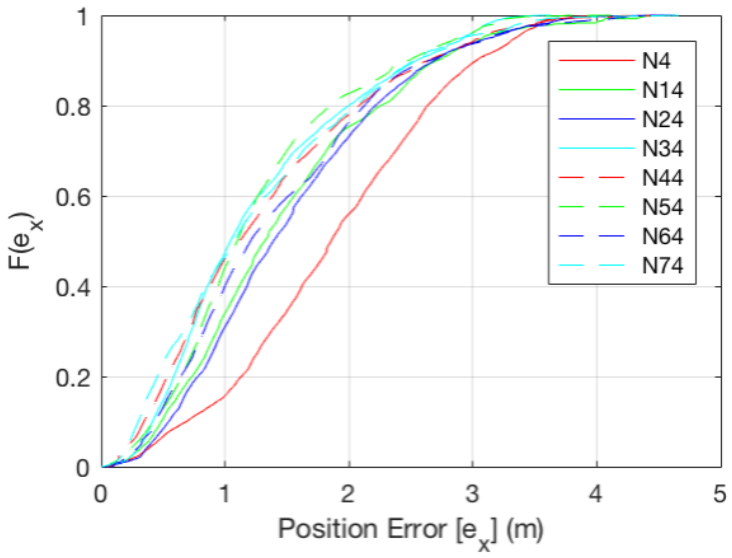


Figure 3–6 Positioning Error CDF

3.4 Summary

This chapter presented the Parametric Grid Sparse Gaussian Process, a Gaussian process sparsification algorithm for efficient feature map representation. The objectives of the algorithm are twofold: to increase the computation efficiency of the GP model, and to decrease the storage and bandwidth requirements of the models. The first objective is accomplished by reducing the total number of points in the training dataset. The second objective is achieved by fixing the locations of the pseudo-inputs on a parametric grid. So, instead of storing the explicit location of the training points, or the pseudo-inputs, the points are indexed relative to the parametric grid. Using PGSGP can reduce the storage requirements by at least 50% while maintaining a reasonable mapping and positioning accuracy.

This chapter proposed a light-weight map representation for the environmental features maps using a sparse Gaussian process. This representation allows for efficient map dissemination between nodes or between the nodes and a centralized processing centre, in addition to increasing the speed of any map-related operations.

Chapter Four: **UWB-Based Collaborative Positioning and Localization**

Ultra-wideband (UWB) technology can achieve centimetre ranging accuracy, and it could be used for relative range measurements between nodes. The basic operation principle of the UWB-based localization systems is node trilateration using measured ranges to multiple anchors with known positions. In real-world indoor environments, the range measurements available locally for any given node could be unreliable or insufficient for localization–the reasons for this include anchors occlusion, excessive multipath errors, or poor anchors geometry. Additionally, the nodes are usually asymmetric–in terms of the available sensors onboard, the computational resources, and the power capacity. This asymmetry affects the positioning performance of the weaker, less resourceful nodes. Collaboration between different nodes is a key element to enable successful nodes localization when the measurements are insufficient or when the local resources are limited. In operating scenarios, where there is a group of nodes in the same physical proximity, and some of these nodes can position itself with higher relative accuracy, using relative measurements can augment the standalone observations of each node and improve the positioning accuracy of the ensemble. Using relative range measurements introduces additional constraint to the position estimation algorithm, which can improve the positioning accuracy of the collaborating nodes. Ultra-wideband (UWB) ranging devices can distinguish between different events with a precise temporal resolution, thus mitigate some multipath errors, thanks to the large channel bandwidth. This makes the UWB-based ranging systems capable of achieving centimetre-level accuracy in ideal operating conditions.

This chapter describes the software and the hardware components of locSpeck, a collaborative positioning system based on a UWB ad-hoc and dynamic network. The ranging measurement component of the system is a custom-built ranging device, based on the DW1000 UWB

transceiver chip from Decawave, which could be paired to a set of smart devices equipped with asymmetric sensors. The described system supports online addition and removal of nodes, and dynamic node role assignment, either as an anchor or as a rover. The performance of the system is evaluated experimentally by using four locSpeck nodes.

The chapter starts by introducing the UWB technology, the different positioning techniques used in conjunction with UWB range measurements, and the different network architectures used for range-based positioning applications. The following section introduces the UWB-based positioning system. This section starts by describing the ad-hoc network architecture employed by the system. Then, it provides the implementation details of the hardware and software components of the ranging device–the basic building block of the proposed system. The structure of the ranging message and the medium access protocol are described later in this section. The following section provides an overview of the dynamic motion model of the nodes, along with the measurements update model. The results are presented in section 4.4, including the timing of the range measurements messages, the theoretical limits of the medium access control protocol performance, and the positioning and localization accuracy of the system. The chapter is concluded with a brief discussion.

4.1 Background

4.1.1 Ultra-Wideband Radios: A Very Brief Introduction

The original idea behind the ultra-wideband (UWB) technology is to use impulses or very short pulses to spread the data over a wideband [139], [140]. The same concept, used earlier for radar applications, formed the basis for the ultra-wideband radio communication [141]. Ultra-wideband radios are strong candidates for indoor communication systems, due to the robustness

of the UWB signal and its immunity to multipath fading [142]–a desired characteristic for a radio signal propagating in the cluttered indoor environments.

UWB systems can have low-complexity and low-cost implementations. UWB systems can lower the possibility of interfering with the already existing wireless infrastructure. They are resilient to multipath propagation since the very short pulses make it easier to distinguish between the different components of the reflected signal. Moreover, they are capable of providing high resolution, required for positioning and tracking applications [139].

The breakthrough of ultra-wideband into the mainstream of consumer and commercial applications occurred on 2002, when the Federal Communications Commission (FCC) allowed the unlicensed use of the UWB-based communication and measurement devices in the 3.1-10.6 GHz frequency band [143], with power limitation of -41.3 dBm/MHz [144]. The FCC report defines the UWB signal bandwidth to be at least 500 MHz or having a relative bandwidth of 20%. On February 2007, the Commission of European Communities followed suit by publishing their regulatory framework for the UWB radio technology [145].

The IEEE standardization effort of the UWB systems resulted in two different implementations, the IEEE 802.15.3 and the IEEE 802.15.4–the former is targeting the small-range high-speed applications, and the latter is targeting the low-data-rate applications. The IEEE 802.15.4-2011 standard defines the physical layer for UWB-based systems for low-power and low-complexity applications [146], which makes it a perfect candidate for wireless sensor networks [147].

4.1.2 UWB-Based Positioning Systems

Ultra-wideband positioning and localization systems have been used in different scenarios that require centimetre ranging accuracy with constraints on the cost and the power of the ranging devices. The applications of UWB-based localization systems include first responders in

emergency situations, assets tracking and monitoring, medical and wellness applications, security and access control, nodes localization in wireless sensor networks, and for military applications [147]–[150].

Positioning using ultra-wideband radios can be performed using different techniques: Angle-of-Arrival (AOA), Received Signal Strength (RSS), Time-Difference-of-Arrival (TDOA), or Time-of-Arrival (TOA) or Time-of-Flight (TOF) [148]. The angle-of-arrival based systems are complex since they require antenna-arrays on the same node, which increases the cost and the complexity of the implementation. The time-based approaches are more suited to the UWB systems as the high-bandwidth of the signal can provide very fine spatial resolution in addition to increasing its immunity to multipath effects.

The Cramér–Rao bound for the time-of-arrival ranging accuracy using the IEEE 802.15.4a–the predecessor to the IEEE 802.15.4-2011–under single-path additive white Gaussian noise (AWGN) channel model, can be expressed as:

$$\sigma_R \geq \frac{c}{2\pi \times \beta \times \sqrt{2(SNR)}}\ , \tag{4.1}$$

where σ_R is the standard deviation of the range estimation R, c is the speed of light, β is the effective bandwidth, and SNR is the signal-to-noise ratio [148], [151]. Setting $\beta = 500$ MHz and $SNR = 10$, the standard deviation of the range estimation $\sigma_R = 2$ cm [21].

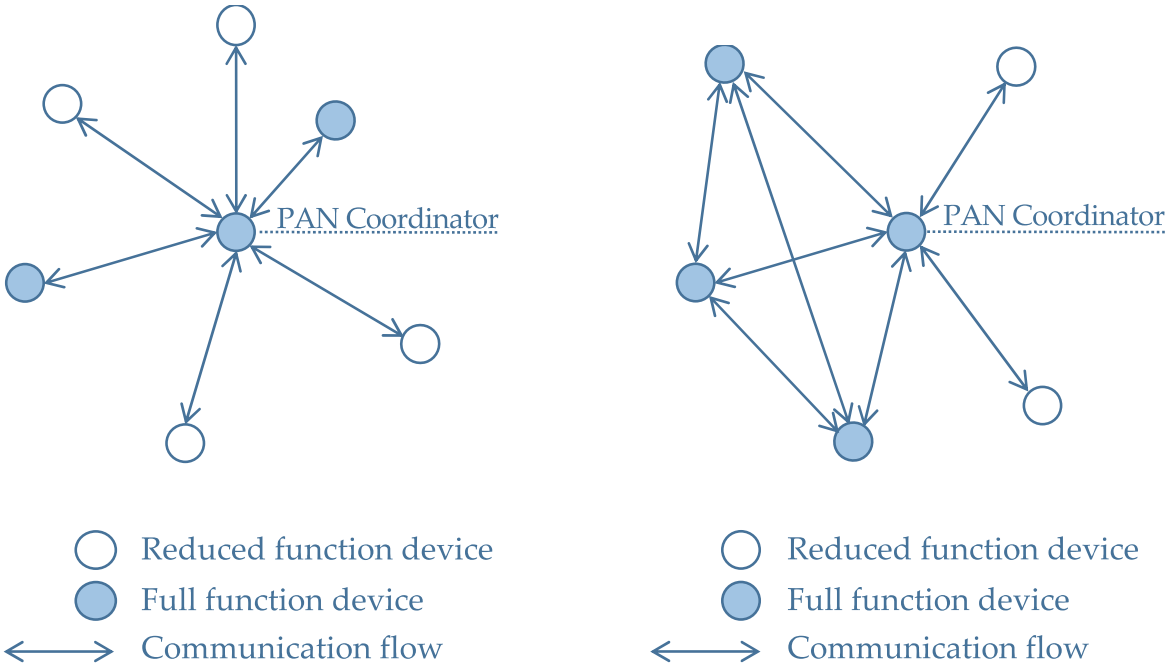
The RSS of the UWB signal is less susceptible to small scale fading compared to narrow-band signals [152], as a result of the large bandwidth of the UWB signals. However, the ranging accuracy achievable using RSS methods decreases with distance [153], making the achievable accuracy less than the accuracy obtained using the time-delay methods. The accuracy of the range measurement using RSS techniques can be expressed as:

$$\sigma_R \geq \frac{ln\,10}{10}\frac{\sigma_{sh}}{n_p}\,d\,, \tag{4.2}$$

where d is the distance between the two nodes, n_p is the path loss factor, and σ_{sh} is the standard deviation of the zero-mean Gaussian random variable representing the log-normal channel shadowing effect [148]. This accuracy could be sufficient for certain applications that do not require a centimetre ranging accuracy. RSS localization relies on two techniques. First, the range estimation knowing the path-loss-model (PLM) and the channel state information (CSI) [152], [154]. Second, signal strength fingerprinting [155] which requires a learning phase to collect the RSS fingerprints along with a set of reference points, then during the localization phase actual RSS value is compared to the previously generated fingerprint database to estimate the location of the UWB receiver in real-time.

4.1.3 UWB Network Architecture

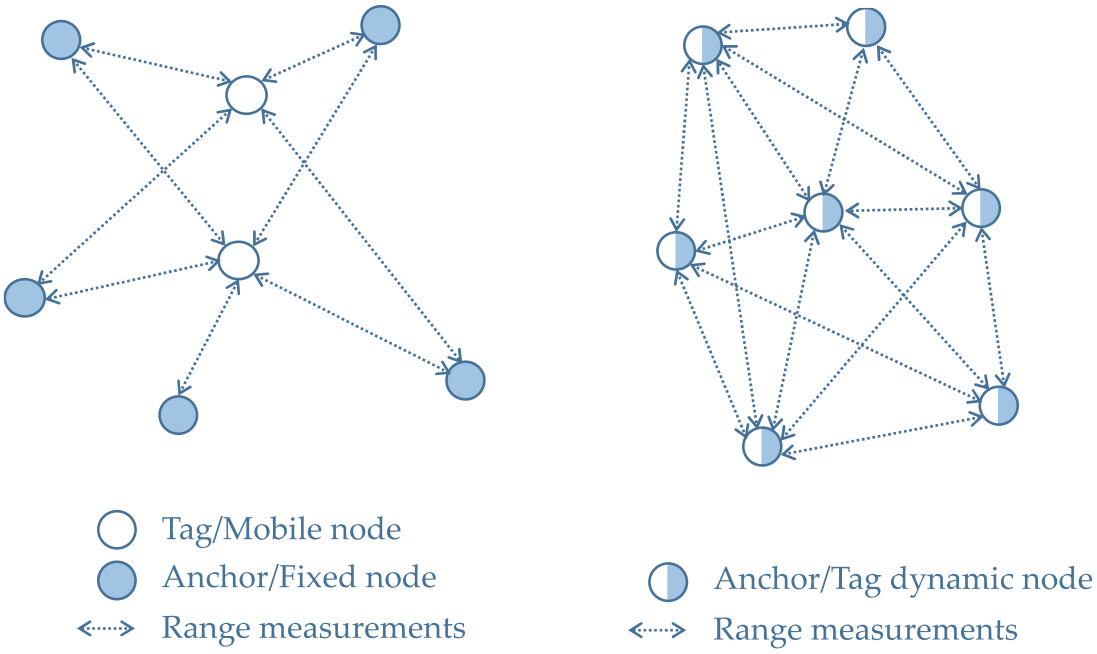
The IEEE 802.15.4-2011 standard supports two network architectures for UWB devices in the context of the Personal Area Network (PAN) [146, p. 15], as shown in Figure 4–1: the star network architecture, and the peer-to-peer network architecture. A typical network configuration of UWB-based positioning system comprises a set of fixed anchors and one or more mobile nodes. The position and number of the anchors are usually known beforehand, and the location of the mobile nodes is estimated using the range measured between those nodes and the anchors. An example of a commercial system for positioning, based on the DW1000 [156] UWB radio from Decawave, is the Pozyx system [157], [158]. The Pozyx system supports a fixed network architecture [159], such that it requires the prior knowledge of the number of the mobile nodes and the fixed nodes in the network–referred to as tags and anchors, respectively. The position of tags is calculated sequentially using a Time-Division-Multiple-Access (TDMA) approach to



(a) (b)

Figure 4–1 IEEE 802.15.4-2011 supported network topologies; (a) Star network topology, (b) Peer-to-peer network topology

minimize the interference possibility between the different tags. In the Posyz system, the positioning and ranging procedures are effectively initiated from a single device in the entire network, which controls the other nodes or tags remotely according to the list of available nodes. Another aspect of network architecture is the design of Medium Access Control (MAC) protocols, which has been studied extensively in the context of wireless sensor networks (WSNs). The design of MAC protocol for sensor networks is guided by the operational goals of the network, and it is usually an attempt to balance between two conflicting goals: achieving high-throughput and maintaining energy-efficient operation. Several recent survey articles provide an insight into the taxonomy of MAC protocols and to the development of different MAC protocols [160]–[164]. The MAC protocols for WSNs can be categorized into



(a) (b)

Figure 4–2 UWB-based network architecture for ranging and positioning applications; (a) Fixed role network, (b) Dynamic role network

asynchronous, asynchronous, frame-slotted, and multi-channel protocols. Although the IEEE 802.15.4-2011 standard includes a definition to the MAC layer, not every UWB radio chip implements the full standard. For example, the Decawave DW1000 does not implement the MAC layer from the standard and leaves this task to the host system [156].

4.2 Ad-Hoc UWB-Based Positioning System

This section describes the hardware and the software aspects of the proposed ad-hoc UWB-based positioning system. The system supports ad-hoc network architecture, run-time inclusion and removal of nodes, and dynamic role assignment to nodes. The section starts by describing the supported network structure, then follows by describing the hardware and software underpinnings of the system. The details of the ranging messages and the medium access protocol are discussed later in this section.

4.2.1 Ad-Hoc Network Structure

The nodes in the proposed system are connected by ad-hoc network architecture, with no fixed structure nor fixed roles. Since the main objective of the network is to perform ranging measurements between neighbouring nodes within the line-of-sight of their UWB receivers, the ad-hoc network described here has some different characteristics from an ad-hoc network used for communication application [165]. A summary of the network characteristics supported by the proposed positioning system is presented in the following points:

- Flat network topology: the network is composed of symmetric nodes in terms of its communication capability, which means that each node can initiate a ranging request or respond to such request from other nodes. Besides, there are no coordinating nodes–as opposed to the peer-to-peer network architecture described in the IEEE 802.15.4-2011 standard [166], [146]. The sensing and computational capabilities of the nodes can still be asymmetric.
- Single-hop network: the nodes are only interested in exchanging ranging messages with neighbouring nodes, within the range of their UWB receivers.
- Energy conservation: after either a failed or successful ranging exchange attempt, the radio chip goes into sleep mode for a defined period before retrying engaging in new ranging sequence exchange.
- Flexibility: nodes can enter and exit the network in real-time, without the need to reconfigure or even notify the existing nodes.

These characteristics emphasize the main objective of the network: ranging and positioning. The nodes are identical in terms of ranging capabilities, low-power, and could be spread over a large

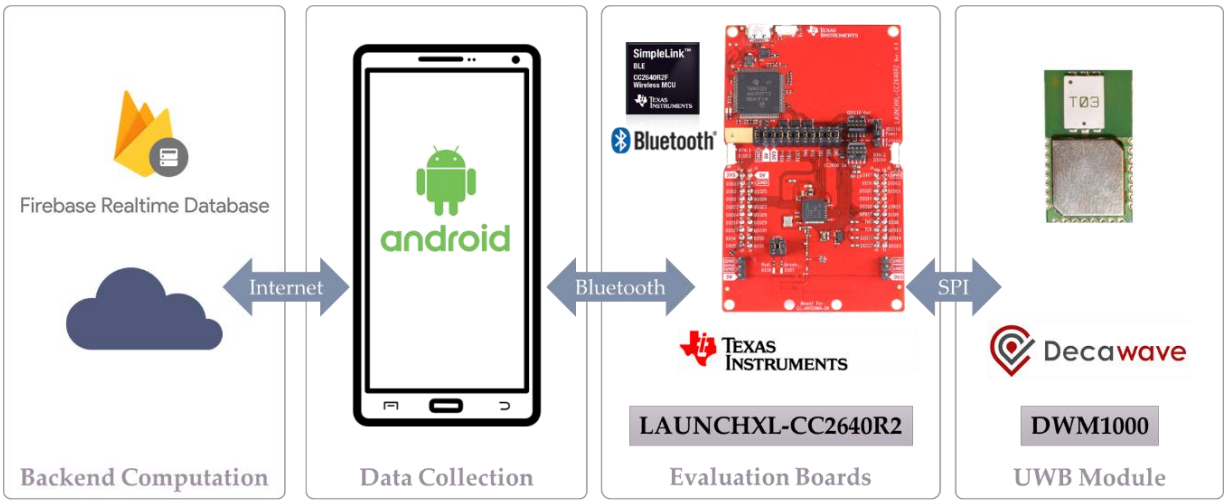


Figure 4–3 Hardware platform overview

physical space. The differences between the ad-hoc network topology and the fixed-role network topology for ranging and positioning applications can be seen in Figure 4–2.

4.2.2 Node Hardware Architecture

The dynamic nodes are the building blocks of the proposed ad-hoc UWB positioning system. Each node is composed of the ranging device and an associated smartphone. An overview of the hardware architecture of the nodes is shown in Figure 4–3. The ranging devices are based on commercial off-the-shelf components. The UWB radio module used is the DWM1000, which is based on the Decawave DW1000 radio and equipped with an onboard chip antenna [167]. The DWM1000 is attached to a CC2640R2 LaunchPad kit from TI [168], which hosts a CC2640R2F wireless microcontroller unit (MCU), enabling Bluetooth low-energy (BLE) communication with the attached DW1000 device. The UWB module and the MCU are connected through a Serial Peripheral Interface (SPI), allowing the host MCU to configure the UWB radio, initiate range measurements, and obtain data and status information from the radio. The UWB module and the BLE evaluation board from TI are enclosed in a custom-built plastic case, which hosts the

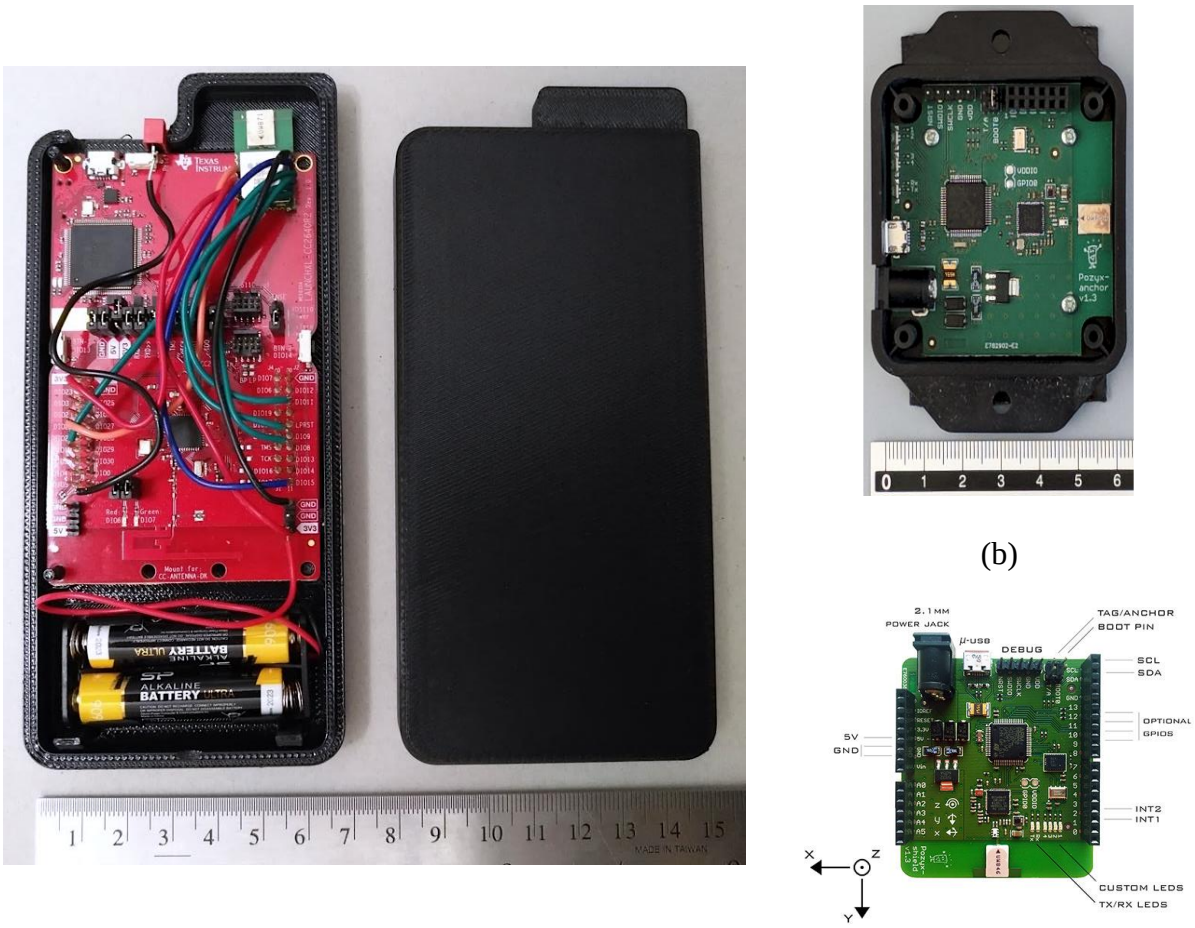
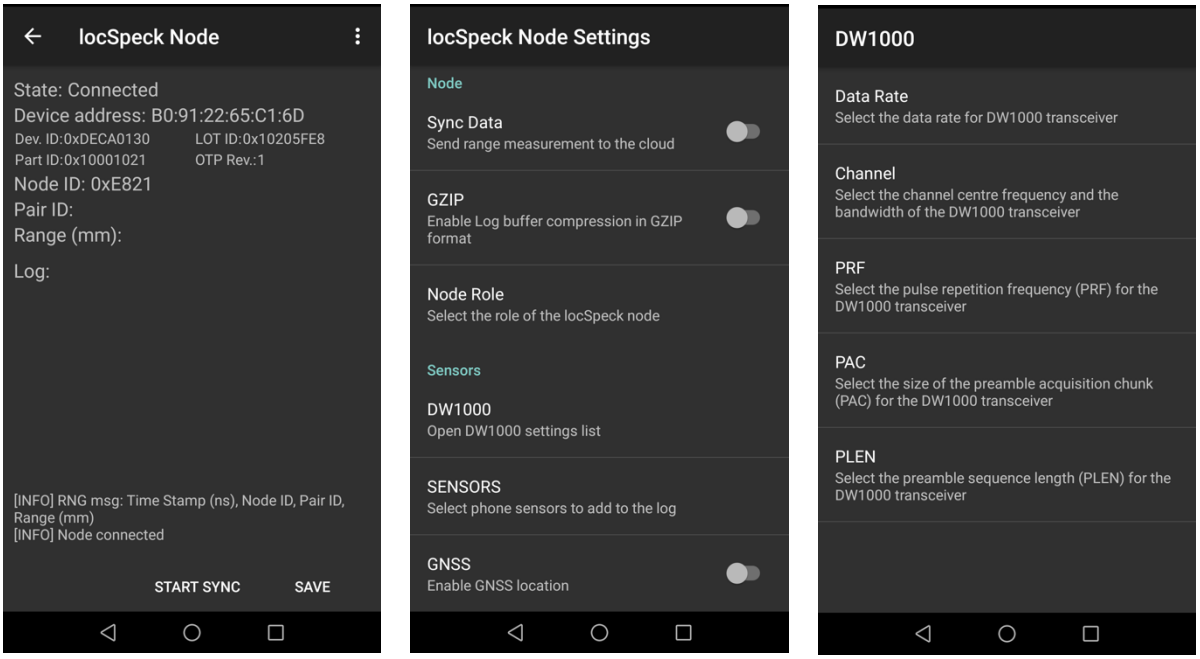


Figure 4–4 The locSpeck Ranging devices; (a) Ranging devices for the proposed UWB-based positioning system, (b) Pozyx positioning system–anchor node, (c) Pozyx positioning system–tag node

batteries and the power switch, as shown in Figure 4–4 (a). For comparison, the Pozyx system components–the anchor and the tag nodes–are shown in Figure 4–4 (b), (c).

The host smartphone connects to the ranging device through a Bluetooth Low Energy (BLE) interface; the smartphone also runs the sensor logging application, shown in Figure 4–5. The ranging device firmware and the logging application communicates through the Generic Attribute Profile (GATT), which is part of the Bluetooth low-energy protocol stack, responsible



(a) (b) (c)

Figure 4–5 The locSpeck logging application on Android; (a) Data logging screen, (b) locSpeck node settings screen, (c) Decawave DW1000 settings screen

for the actual data exchange between any two connected devices. The GATT stores and passes the data as a set of characteristics, stored in the memory of the BLE device. The GATT profile implemented by the locSpeck ranging device is summarized in Table 4–1. The locSpeck logging application can identify the attached UWB chip, read and change the configuration of the UWB chip, and log the range and the paired node ID. The locSpeck logging application can also collect data from the smartphone sensors (e.g., accelerometer, gyroscope, magnetometer, Wi-Fi RSSI, barometers), and it can record the GNSS position information.

The locSpeck system allows all nodes to send their ranging measurements to a remote real-time database, enabling the implementation of a centralized and high-level positioning and mapping algorithms in the cloud. However, for the rest of this work, the positioning algorithm was

implemented on a desktop computer in a post-processing mode, and the locSpeck nodes and applications are used for logging data only.

Table 4–1 The locSpeck BLE GATT profile.

Characteristic	Properties	Size (bit)	Description
DEVID	R	32	DW1000 Device ID
PARTID	R	32	DW1000 Lot ID
OTPREV	R	8	DW1000 OTP Revision
Range	R (Notify)	32	Range measured
Pair ID	R (Notify)	16	The ID of the paired node
CONF	R/W	16	Node settings
Node ID	R	16	Node ID

4.2.3 Range Measurement Messages

The ability of UWB systems to provide accurate ranging is the key motivation to use them for indoor positioning and localization. The relative range measurements between two UWB transceivers can be achieved using delay or time-based methods, angle-of-arrival methods, or received signal strength methods [148]. As shown earlier in Equations (4.1) and (4.2), the Cramér-Rao lower bound for range measurements using the received signal strength increases with the distance, lowering the achievable positioning accuracy. The angle-of-arrival method requires antenna arrays to distinguish the phase of each of the incident radio rays to calculate the relative angle between nodes. The superior theoretical performance and the simpler implementation makes the time-based ranging solution an attractive option for commercial positioning solutions such as receivers from Decawave, BeSpoon, and Ubisense [169], though the Ubisense system supports the angle-of-arrival (AOA) measurements as well.

Before proceeding to describe the range measurement algorithm implemented by the locSpeck node, two time-based ranging methods will be briefly discussed: the time-difference-of-arrival (TDOA) and the time-of-flight (TOF) methods. In TDOA-based systems, the tag or the mobile

node sends a periodic message which is received by the surrounding anchors. The internal clocks of the anchors must be synchronized so that the anchors can compare the arrival time of the tag message using the same time reference. In TOF-based systems, two-way communication between the neighbouring nodes is required to calculate the time-of-flight without the need for synchronizing different nodes. The time-of-flight is converted to a range measurement by multiplying it by the speed of light. The DW1000 chip can implement both methods. However, the locSpeck system implements the time-of-flight method for range calculation.

Figure 4–6 shows the message exchange for two different time-of-flight ranging techniques utilizing the Decawave DW1000 chip. Although the DW1000 chip does not implement the top-level ranging technique, the chip provides means to precisely control the messages exchange and to accurately time-stamp the transmission of the messages [156]. The host system–in the case of locSpeck, the ARM Cortex-M3 MCU embedded on the TI CC2640R2 chip–is responsible for implementing the ranging algorithm.

Figure 4–6 (a) shows the messages exchanged between two nodes for the single-sided two-way ranging. The propagation time can be calculated using Equation (4.3):

$$T_{prop} = \frac{1}{2}\left(T_{round} - T_{reply}\right), \tag{4.3}$$

where T_{round} and T_{reply} are the round-trip time and reply time, respectively. Each time quantity is measured on device A and device B using their local clocks, alleviating the need to synchronize the nodes. This method represents a simple approach to calculating the range, with the exchange of two messages only. However, the drawback of the single-sided ranging method is that the error in the range measurements increases as the reply time increases. The reason for this error is attributed to the small clock offset from its nominal value in the oscillator of each

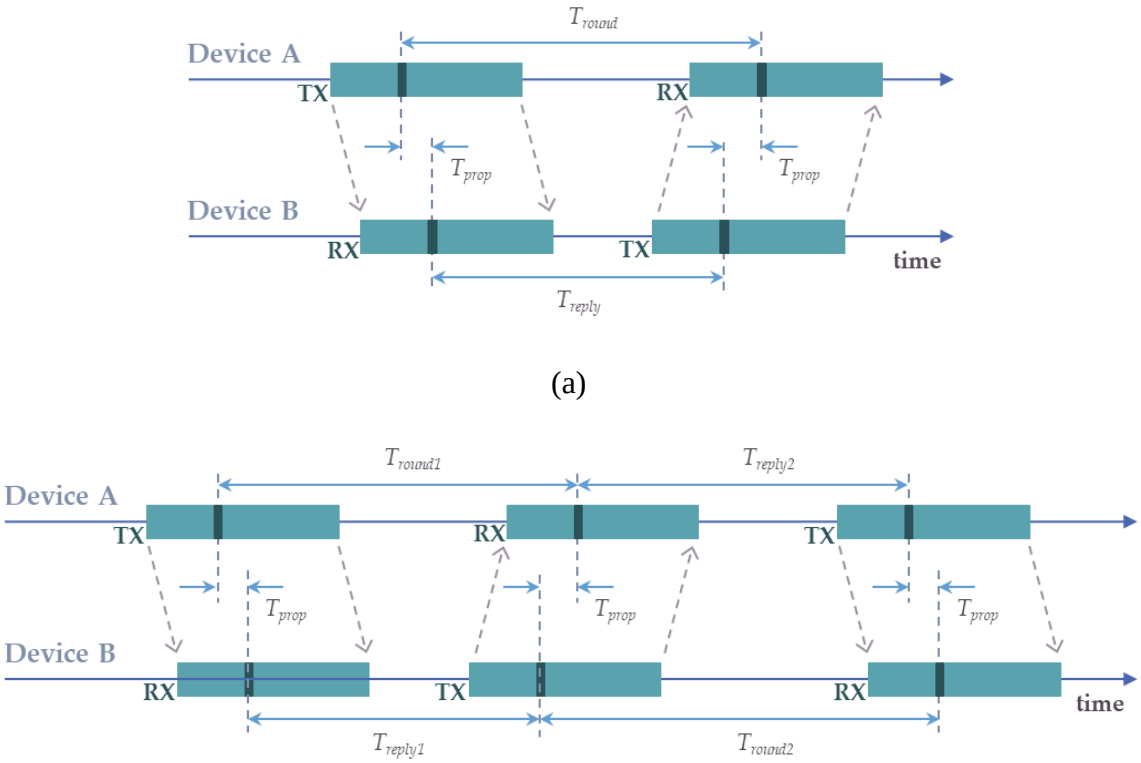
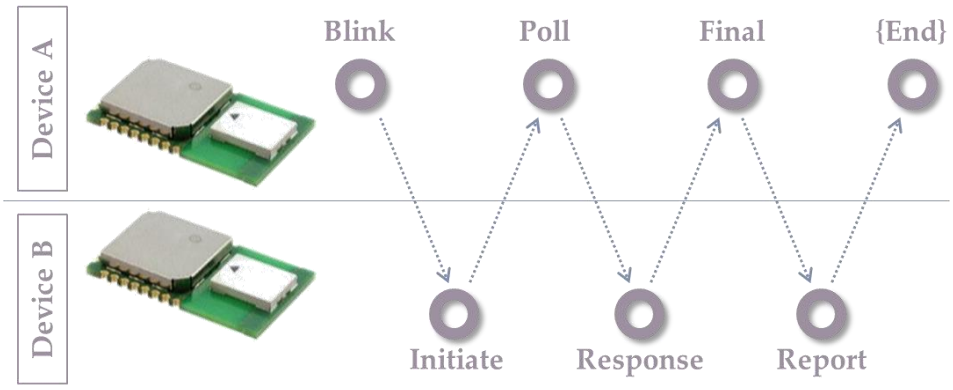


Figure 4–6 Two-way ranging frame sequence; (a) Single-sided two-way ranging, (b) Asymmetric double-sided two-way ranging.

chip. Figure 4–6 (b) shows the asymmetric double-sided two-way ranging method in which the ranging exchange requires three messages, where two round trips are combined to calculate the propagation time, reducing the ranging errors. The asymmetry in this exchange is manifested by the fact that the reply time of both nodes is not equal. The propagation time for the asymmetric two-way ranging can be calculated using Equation (4.4):

$$T_{prop} = \frac{T_{round1} \times T_{round2} - T_{reply1} \times T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}} . \quad (4.4)$$

The propagation time calculated in Equation (4.4) ensures that the error due to clock offset is minimized, compared to the single-sided method.



(a) (b)

Figure 4–7 Ranging message structure; (a) Decawave DWM1000 modules, based on DW1000 UWB radio chip, (b) Ranging messages exchanged between two nodes

The locSpeck node implements the double-sided two-way method using three messages for ranging in addition to one final message to share the range between the nodes pair, as shown in Figure 4–7. In this example, the messages exchange is expressed as follows:

- Device A begins the ranging exchange by sending a blink message to any of the surrounding nodes. The purpose of this message is to notify any available nodes that device A is prepared to proceed with the range measurement exchange.
- If device B is within the communication range and is listening to the correct UWB channel, it receives the blink frame and replies by sending the range measurement initiation message, using the address of device A.
- Device A receives the initiate message then it sends back a poll message to the other side and records the precise time of sending the poll frame.
- Device B gets the poll message and stamps the arrival time. Then, device B sends a response message to device A and record the reply time (T_{reply1}).

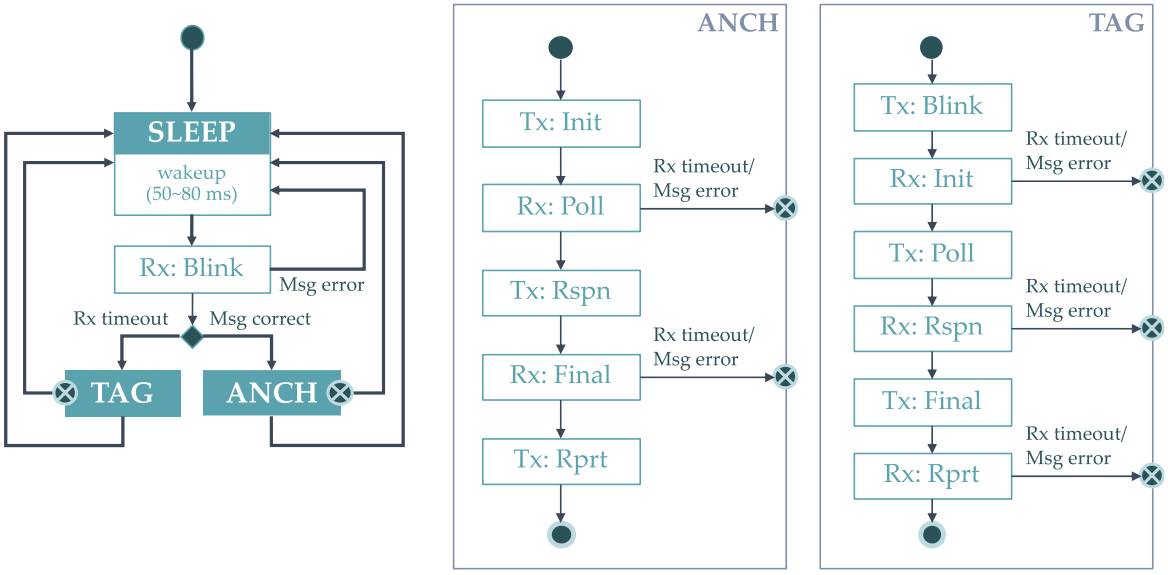


Figure 4–8 locSpeck medium access protocol

- Device A gets the response frame and saves the arrival time stamp, and then calculates the first round-trip time (T_{round1}). After the second reply time (T_{reply2}), device A sends the final message.
- Device B receives the final frame and record the round-trip time (T_{round2}). Using Equation (4.4), device B calculates the propagation time (T_{prop}), and the range consequently. Finally, device B sends the propagation time back to device A.

By the end of the messages exchange, the exact range value is available at both devices for further processing. The positioning algorithm uses the range information along with other sensors readings to update the position state of the nodes.

4.2.4 Medium Access Protocol

The locSpeck nodes are designed to operate in an unpredictable environment–unpredictable in terms of the number of the surrounding nodes and in terms of the possible structure of the network formed using these nodes. The nodes should support rapid deployment with no or minimum effort from the operator. As a result, locSpeck nodes implement a simple and light-

weight random-access medium access protocol based on the pure ALOHA protocol [170], [171]. Figure 4–8 shows an overview of the medium access protocol implemented by the locSpeck nodes.

After powering up locSpeck nodes, they start in the sleep state. Each node sleeps for a random duration between 50 ms and 80 ms. When the sleep duration elapses, nodes wake-up and power-up its receiver, waiting for any incoming frames. If a node receives a complete frame, it decodes the frame and checks if the incoming frame is a blink message. In the case of receiving a blink message, the node operates as an anchor. If the node receives a complete frame, and it is not a blink message, it concludes that the channel is currently occupied by the ranging sequence of another pair of nodes, and goes back to the sleep state, so it would not interfere with the current exchange. If no frames are received and the listening period elapses, the receiver timeout flag is asserted. In this case, the node assumes that the channel is free, and it switches to the tag mode and starts sending a blink message to any active node.

While at the anchor or the tag states, each node sends or receive a sequence of messages. If any of the received messages does not match the expected message at that stage, the node will switch back to the sleep mode. Also, the nodes will go to the sleep mode, if there is any problem with messages transmission or reception, such as a receiver timeout, or any other problem related to the radio interface. Once a ranging sequence is completed successfully, the node saves the range and the node ID of the collaborator. Finally, each node sends a notification to the host smartphone before returning to the sleep mode for another random duration.

The locSpeck medium access protocol assumes that all the nodes use the same UWB channel and the same preamble code for both communications and for ranging. The DW1000 supports the use of 6 RF channels out of the 16 channels defined in the IEEE 802.15.4-2011 standard.

Each node operating in the channel is assigned a preamble code from a set of 2 or 4 possible codes. The exact values of the preamble sequence are defined by the standard and are selected to ensure minimum cross-correlation between different codes. Assigning different nodes to different channels and assigning different preamble codes to the nodes operating in the same channel can increase the number of nodes operating in close proximity [172]. Although the multiple-channel and multiple-preamble approach can increase the effective number of nodes, its use was not considered in this work since it would increase the complexity of the system. To use this approach, more functionality is needed on top the current protocol to scan different channels and preambles and to keep track of nodes in each channel-preamble configuration.

4.3 Collaborative Positioning Algorithm

This section describes the positioning algorithm implemented by the locSpeck system. It highlights the dynamic motion model of the nodes, along with the measurements update model. The locSpeck nodes use two different measurement update models: relative range measurement updates and Wi-Fi fingerprinting updates. The locSpeck node positioning accuracy is evaluated and compared to the position estimates of the Pozyx system. The position estimation process for the locSpec nodes is performed using a particle filter, where the details of the filter implementation are discussed in the rest of this section.

4.3.1 Dynamic Motion Model

The dynamic model used in the locSpeck positioning filter is a random walk motion model while the state vector comprises the 2-D position of each node. The use of the other available sensors, such as the accelerometer, the gyroscope, or the magnetometer, can help improve the positioning performance. However, to keep the model simple, the use of these sensors is forgone for the evaluation of the ranging performance. The dynamic model is described by Equation (4.5):

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + u_{k+1}, \tag{4.5}$$

where x_{k+1}, y_{k+1} are the position of the nodes, and u_{k+1} is a normally distributed, zero-mean, white noise, with variance Q_{k+1}, i.e. $w_{k+1} \sim \mathcal{N}(0, Q_{k+1})$.

4.3.2 Measurements Update

The weights of the particles in the filter are updated using the relative-range, and the Wi-Fi received signal strength indicator (RSSI) measurements, using Equation (4.6):

$$\widetilde{w}_{k+1}^{i} \propto w_k^i \times p\left(z_{k+1} \middle| x_{k+1}^i, y_{k+1}^i\right), \tag{4.6}$$

where w_k^i is the current weight of particle i, $\widetilde{w}_{k+1}^{i}$ is the updated weight, z_{k+1} is the value of the observation, and $p\left(z_{k+1} \middle| x_{k+1}^i, y_{k+1}^i\right)$ is the likelihood of observing z_{k+1} at the location defined by x_{k+1}^i and y_{k+1}^i. Equation (4.6) is a simplified version of the weights update equation, in which the proposal density is the state transition model, defined by Equation (4.5). The details of the measurement likelihood equations for different measurement updates equations are presented in the following few lines.

4.3.2.1 Range Measurement Update

The measurement update using relative-range between a pair of collaborating nodes involves two pieces of information: the relative-range and the position of the collaborating node. The measurement can be described using Equation (4.7):

$$h_{k+1}^{i,R}\left(x_{k+1}^i, y_{k+1}^i\right) = \sqrt{(x_{k+1} - x_0)^2 + (y_{k+1} - y_0)^2}, \tag{4.7}$$

where $h_{k+1}^{R}(\cdot)$ is the range measurement estimate at time step $k+1$; x_{k+1}^i, y_{k+1}^i are the position of the particle i; and x_0, y_0 are the coordinates of the collaborating node. The range estimate is used to evaluate the likelihood of the range measured given the position of each particle, $p(z_{k+1}^R | x_{k+1}^i, y_{k+1}^i)$, as described by Equation (4.8).

$$p\left(z_{k+1}^{R}\middle|x_{k+1}^{i},y_{k+1}^{i}\right)=(2\pi\sigma_{R}^{2})^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\frac{\left(z_{k+1}^{R}-h_{k+1}^{i,R}\right)^{2}}{\sigma_{R}^{2}}\right),\tag{4.8}$$

where z_{k+1}^{R} is the actual range measurement, which is modelled as a Gaussian random variable with mean equals to the measured range and covariance of σ_{R}^{2}.

4.3.2.2 Wi-Fi RSSI Fingerprint Update

The Wi-Fi RSSI fingerprinting method used in this work uses a Gaussian process model to represent the RSSI map [173]. The Gaussian process model is fully defined by a mean function and a covariance function, μ_{wifi} and σ_{wifi}^{2} respectively [118]. Using the mean and covariance function, the likelihood of observing a certain set of Wi-Fi RSSI values can be described using Equation (4.9):

$$p\left(z_{k+1}^{wifi}\middle|x_{k+1}^{i},y_{k+1}^{i}\right)=\det\left(2\pi\Sigma_{wifi}\right)^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\left(z_{k+1}^{wifi}-M_{wifi}\right)^{T}\Sigma_{wifi}^{-1}\left(z_{k+1}^{wifi}-M_{wifi}\right)\right)\tag{4.9}$$

where z_{k+1}^{wifi} is a vector with all the observed RSSI values at time step $k+1$, M_{wifi} is a vector of the mean function values for each observed Wi-Fi access point (AP), i.e. $M_{wifi}=\left[\mu_{wifi}^{1},\dots,\mu_{wifi}^{N}\right]^{T}$, and Σ_{wifi} is the observations covariance matrix, which is a diagonal matrix with each element in the diagonal represents the covariance value for each observed AP, i.e. $\Sigma_{wifi}=diag\left(\left[\left(\sigma_{wifi}^{1}\right)^{2},\dots,\left(\sigma_{wifi}^{N}\right)^{2}\right]\right)$. The mean and variance functions for each access point, M_{wifi} and Σ_{wifi}, are constructed using a set of training data, $D=\{(x_1,z_1),(x_2,z_2),\dots,(x_N,z_N)\}$, where, x_i is the horizontal position of the training data, i.e. $x_i\in R^2$, and z_i is the RSSI value, measured at the point x_i. More details about the exact implementation of the Wi-Fi RSSI model using Gaussian Process can be found in [173].

4.4 Simulation and Experimental Results

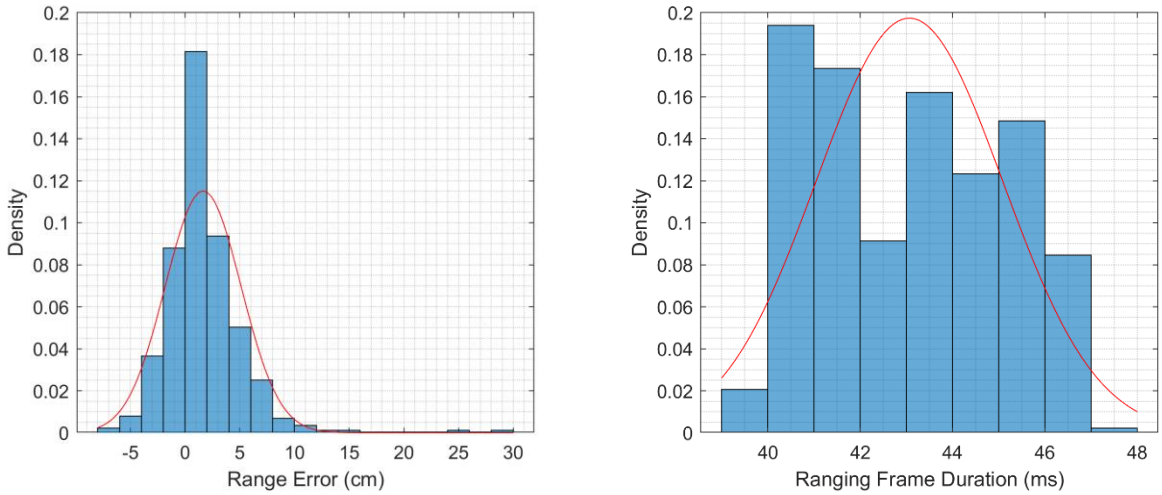
This section summarizes the results for several aspects of the locSpeck positioning system. It highlights the performance of the three building components of the system: the timing of the ranging frames, the efficiency of the medium access protocol, and finally the accuracy of the positioning and localization system using a sample test scenario for the locSpeck system.

4.4.1 Range Measurement Messages Timing

The timing characteristics of the ranging sequences and the individual messages in the sequence were measured and analyzed using DW1000 on-chip high-precision clock for time-stamping the different sequences. To obtain the range and the duration of the ranging frames, two locSpeck devices were placed 60 cm apart, and the firmware was modified to enable logging the frame duration measurements to a computer. The total duration of the ranging frame is affected by the DW1000 chip settings, such as the preamble length and the data rate. The DW1000 chip settings used for the rest of this section are summarized in Table 4–2. Changing the PRF, PLEN, and DR not only affects the frame time, but it can also affect the ranging performance.

Table 4–2 Node settings

Node Setting	Value
Channel number	5
Pulse repetition frequency (PRF)	64 MHz
Preamble length (PLEN)	1024
Data rate (DR)	110 kbps
Range between nodes	60 cm



(a) (b)

Figure 4–9 Range measurement frame statistics; (a) Range error probability density function (pdf), (b) Ranging frame duration pdf

Figure 4–9 shows the histograms of the error of the range measurement values and the total duration of the ranging frames for a sample of approximately 430 range measurements. The size of the data payload of each message in the ranging sequence and the duration of each message is listed in Table 4–3. These values were calculated according to the active DW1000 chip settings and the size of the payload data [174]. The measured ranges using the DW1000 can be affected by noise, uncalibrated bias, and received signal power-dependent biases [175]. The overall ranging frame duration is dominated by the messages sending time and by the processing delays mandated by the firmware implementation and the processor speed. Since the host microcontroller is handling multiple tasks concurrently, the total ranging frame time will account for any other background tasks running during the ranging sequence. The propagation time of the messages from one node to another is 2 ns, which is negligible relative to the transmission and processing delays.

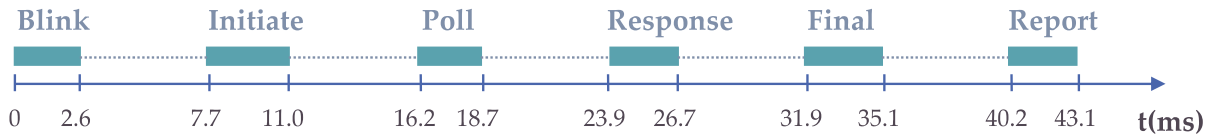


Figure 4–10 Ranging frame–messages exchange timeline

Table 4–3 Ranging messages size and duration

Messages	Size (byte)	Duration (ms)
Blink	12	2.57
Initiate	22	3.32
Poll	12	2.57
Response	16	2.87
Final	20	3.18
Report	16	2.87
Total		17.38

Figure 4–9 shows the distribution of both the range error and the ranging frame duration. At 60 cm separation distance between the two nodes, the mean of the measured range error is 1.6 cm, with 3.5 cm standard deviation. This error can be attributed to multiple factors: residual biases after node range calibration, or measurement setup inaccuracies. The residual bias arises from the fact that the nodes are calibrated using a constant value, whereas the bias is dependent on the power level of the received signal or equivalently on the separation between nodes. The mean and standard deviation of the range error and frame duration are summarized in Table 4–4. Figure 4–10 shows the messages time progression in a ranging sequence. This frame structure is used in evaluating the locSpeck medium access protocol, as discussed in the next section. The messages duration was calculated, as highlighted in Table 4–3, while the average processing time between messages was evaluated using the total ranging frame measurements, captured using the DW1000 precise timing capabilities, and distributed equally among the processing gaps between different messages transmission.

Table 4–4 Range value and frame duration statistics

	Range Error (cm)	Frame Duration (ms)
Mean	1.6	43.1
Standard Deviation	3.5	2.0

4.4.2 Medium Access Protocol Performance

This section examines the performance of the locSpeck medium access protocol in terms of the ranging efficiency as a function of the number of the collaborating nodes. It is expected from a random-access protocol such as ALOHA–the basis of the locSpeck medium access protocol–that the utilization of the medium is reduced, due to collisions between different nodes attempting to initiate ranging sequence at the same time. Using the settings outlined in Table 4–2, the theoretical maximum ranging rate achievable by the locSpeck system is 20 measurements per seconds. The ranging rate is calculated using the average frame duration in addition to 15% of the ranging frame used as a guard interval between different ranging frames.

Figure 4–11 shows the role transition of nodes with time according to the locSpeck medium access protocol, as outlined in Figure 4–8. Figure 4–11 (a) shows the state transition for a node acting as a tag. At time 0, the node wakes up and starts listening for any incoming messages. If no messages were received after 10 ms, the node concludes that the channel is free and is ready for transmission. The node role switches to tag and starts the ranging sequence by sending a blink message. If there is a listening node in the tag proximity, it sends a response, and the ranging sequence will continue for another 43 ms.

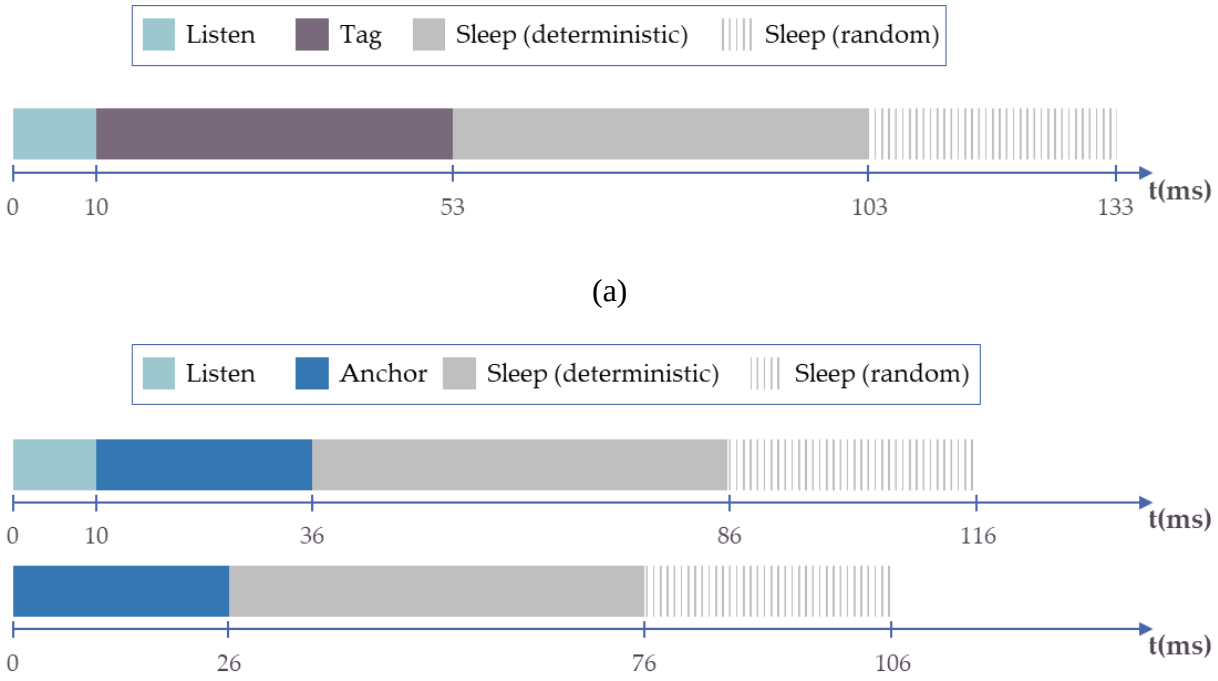


Figure 4–11 Medium access protocol–node role transition; (a) Tag node role, (b) Anchor node role

After finishing the ranging sequence, each node of the pair sleeps for 50 to 80 ms. The sleep interval is a random value which changes every time a node enters the sleep mode. Since the frame duration and the sleep interval is between 103 and 133 ms, the theoretical ranging rate between two locSpeck nodes is 7.5 to 9.7 measurement per seconds. The maximum ranging rate over the channel with the locSpeck medium access protocol is 18.9 measurement per seconds, assuming a new range measurement will start once the active node goes to sleep mode, which occurs every 53 ms. Figure 4–11 (b) shows the timeline when a locSpeck node switches to the anchor role. This switch will occur if the node detects a valid blink message during the listen interval. Since the blink message can be received at any point during the listen interval, the complete frame duration for the node in anchor role ranges between 76 to 116 ms, considering

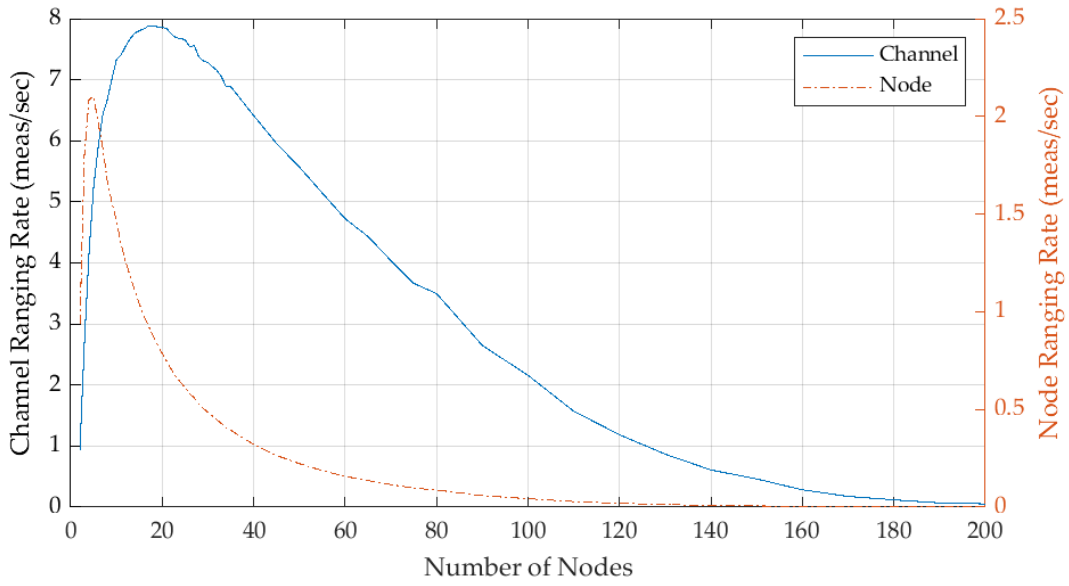


Figure 4–12 Simulated ranging rates results; the average ranging rate over the channel for all nodes and average ranging rate per node

the uncertainty in the first 10 ms listening interval and the uncertainty in the 30 ms sleep random component.

The simulated ranging rates versus the number of collaborating nodes are shown in Figure 4–12 and the maxima of the raging rates are summarized in Table 4–5. The channel ranging rate as a function of the number of nodes is shown as a solid line in Figure 4–12. Under the tested configuration of the system, the maximum utilization of the channel is achieved with 18 nodes, reaching 7.9 measurements per second, which accounts to channel efficiency of 39.5% compared to the theoretical 20 measurements per second. It is worth noting that given fixed time, the ranging rate drops with the number of nodes until it is practically zero. The dotted line in Figure 4–12 shows the average of the ranging rate per node, assuming range measurements are uniformly distributed across the node population. The maximum ranging rate per node is 2.1 measurements per seconds with 5 active nodes achieving 10.4% of the theoretical rate.

Table 4–5 Ranging rate maxima.

	Channel	Node
Number of Nodes	18	5
Ranging Rate (meas/sec)	7.9	2.1

4.4.3 Positioning and Localization Accuracy

4.4.3.1 Experimental Setup

The experiment is done on the main floor of the MacEwan Student Centre at the University of Calgary. The test area is an open lobby covering 300 m^2 (20 m x 15 m), with no obstructions in the middle of the lobby. The duration of the experiment is 24 minutes. The experiment venue is open for the students, and several students were moving around in the vicinity of the experiment. The purpose of selecting this location is to test the system under realistic operational conditions. Figure 4–13 shows the floorplan of the experiment venue.

The experiment setup consists of two separate UWB positioning systems, operating at the same time. The first system is LocSpeck–the ad-hoc system under test–and the other is the Pozyx system, which is used as a reference for comparison. LocSpeck system is composed of four ranging devices. Three of them are fixed at the limits of the test area to act as anchor points, and a fourth device is used as a mobile node. Although these three devices are used as anchors in this experiment, they are functionally identical to the mobile node. The Pozyx system is composed of six anchor nodes, distributed along the perimeter of the test area, in addition to a mobile tag. The area is covered with 25 test points, used to calculate the error in the trajectory. These test points, along with the anchor points from the two systems, were surveyed to establish the ground truth.

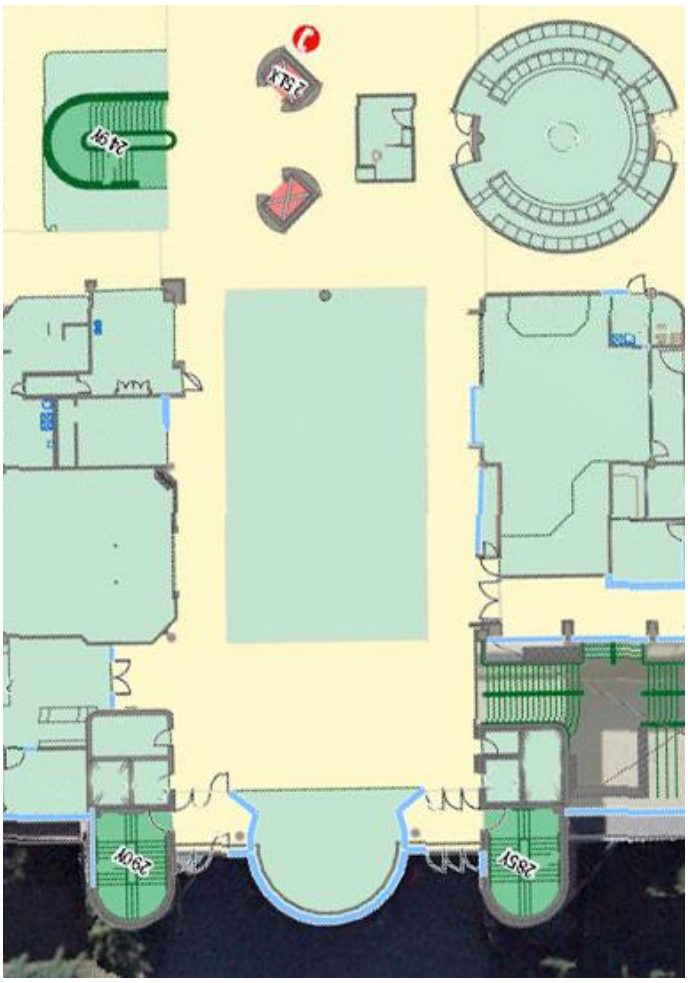


Figure 4–13 Floorplan of the test area

The experiment is divided into two stages: the static self-positioning stage, and the dynamic stage. In the first stage, the measurements between the anchors were collected and used to estimate the positions of the anchors. This process was done for the two systems separately. During the second stage, the two mobile nodes–or tags–are attached to the test subject at the hip level. The dynamic test is performed with the test subject moving between the reference points and stopping at a random reference point briefly. The difference between the actual position of the reference point and the estimated position using the two systems is recorded. The maximum, mean, and the root mean square error of the positioning error is calculated for different system configurations.

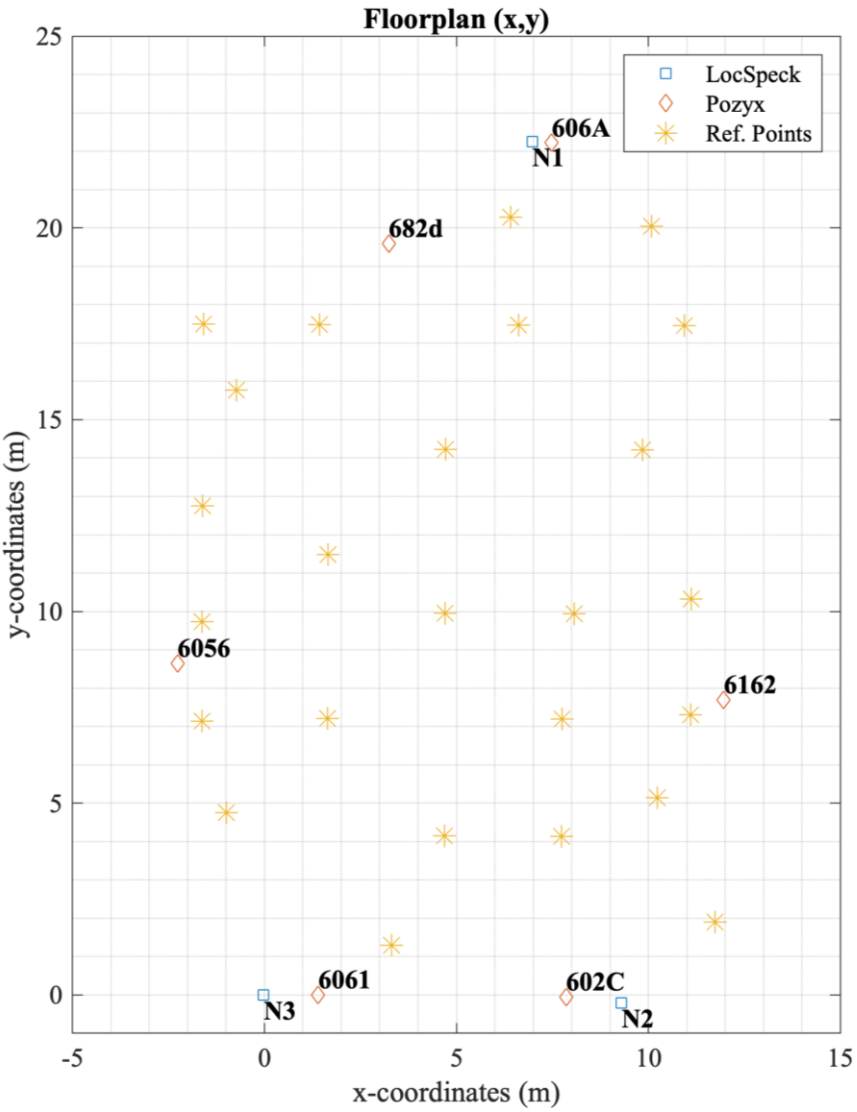


Figure 4–14 Anchors and Reference Points Locations

4.4.3.2 Results and Discussion

4.4.3.2.1 Self-positioning results:

Figure 4–15 and Table 4–6 show the results of the self-positioning step in the experiment. The reported errors are between the estimated positions and actual surveyed positions of the anchors. Figure 4–15 shows the 95% confidence ellipse of the estimated position. Node #3 was fixed to the origin of the 2D mapping frame. Node #2 was initialized to be of the same horizontal line as

the origin. However, nodes #2 and #1 were allowed to move to minimize the square root error for the actual range measurements.

Although the UWB transceivers were calibrated, there is still a residual bias in the range measurements between the different nodes. These range biases result in a bias in the anchor node position estimate. The bias in the range measurement could be explained by the power dependent bias error [175] in the UWB transceiver, which was not compensated. The only error compensated during the calibration process is the constant bias, and it was compensated for short ranges. Table 4–6 shows the errors in millimetre for the horizontal positions of the nodes, along with the Eigenvalues of the scaled covariance matrix, corresponding to the 95% confidence ellipse. Note that the anchor position errors in the self-positioning stage will propagate to the mobile node position errors, during the dynamic positioning stage. The different scenarios used for the dynamic positioning will show the effect of the anchor location errors on the positioning error of the dynamic node.

Table 4–6 Node Positions Error

Node	Error (mm)			Eigenvalues (mm)	
	x	y	$2D$	λ_1	λ_2
1	-154.80	-420.20	447.80	1272.70	8.07
2	-668.60	-22.10	669.00	2.72	229.19
3	-17.60	-1.10	17.70	0.60	0.60

4.4.3.2.2 Dynamic-positioning results:

Figure 4–14 shows the locations of the three LocSpeck nodes, the six Pozyx anchors, and the locations of the reference points used to evaluate the positioning performance of the different combination of anchor nodes. There are 25 unique reference points on the floor. The dynamic node is allowed to move freely between the nodes and stop for a few seconds over one of the

nodes, the position estimate at the reference point is compared with the surveyed position of the node, and the difference is the error in the estimated position. The duration of the test scenario is 24 minutes, and the total distance travelled is approximately 590 m.

Table 4–7 summarizes the results of the dynamic positioning algorithm for different test scenarios. The first column in this table explains the corresponding test scenario briefly. The second and third columns give the number of the LocSpeck and Pozyx nodes used during the scenario. The following three columns show the maximum, mean, and the root-mean-square errors of the horizontal position of the dynamic node. Finally, the last column gives the average range measurements update rate during the scenario.

Following a description of the different scenarios and the corresponding anchor combinations considered for the evaluation of the system:

- Scenario #1: Three LocSpeck anchors, with no assisting Pozyx nodes. This scenario examines the intrinsic performance of the LocSpeck system.
- Scenario #2: Three Pozyx anchors only. The three nodes in this scenario are selected to have similar geometry to that of the LocSpeck nodes in Scenario #1. The purpose of this scenario is to evaluate the performance of the two systems under the same relative conditions.
- Scenario #3: This scenario uses the full set of LocSpeck nodes with two additional Pozyx nodes on the sides to help enhance the geometry of the anchor network.
- Scenario #4: The full set of LocSpeck nodes are used along with four Pozyx anchors.
- Scenario #5: Using the full set of LocSpeck nodes and the full set of Pozyx nodes.
- Scenario #6: Use only the full set of Pozyx nodes. This scenario compares the performance of the Pozyx raw measurements, processed by the dynamic-positioning algorithm of the LocSpeck system to the reference solution, generated by the Pozyx system.

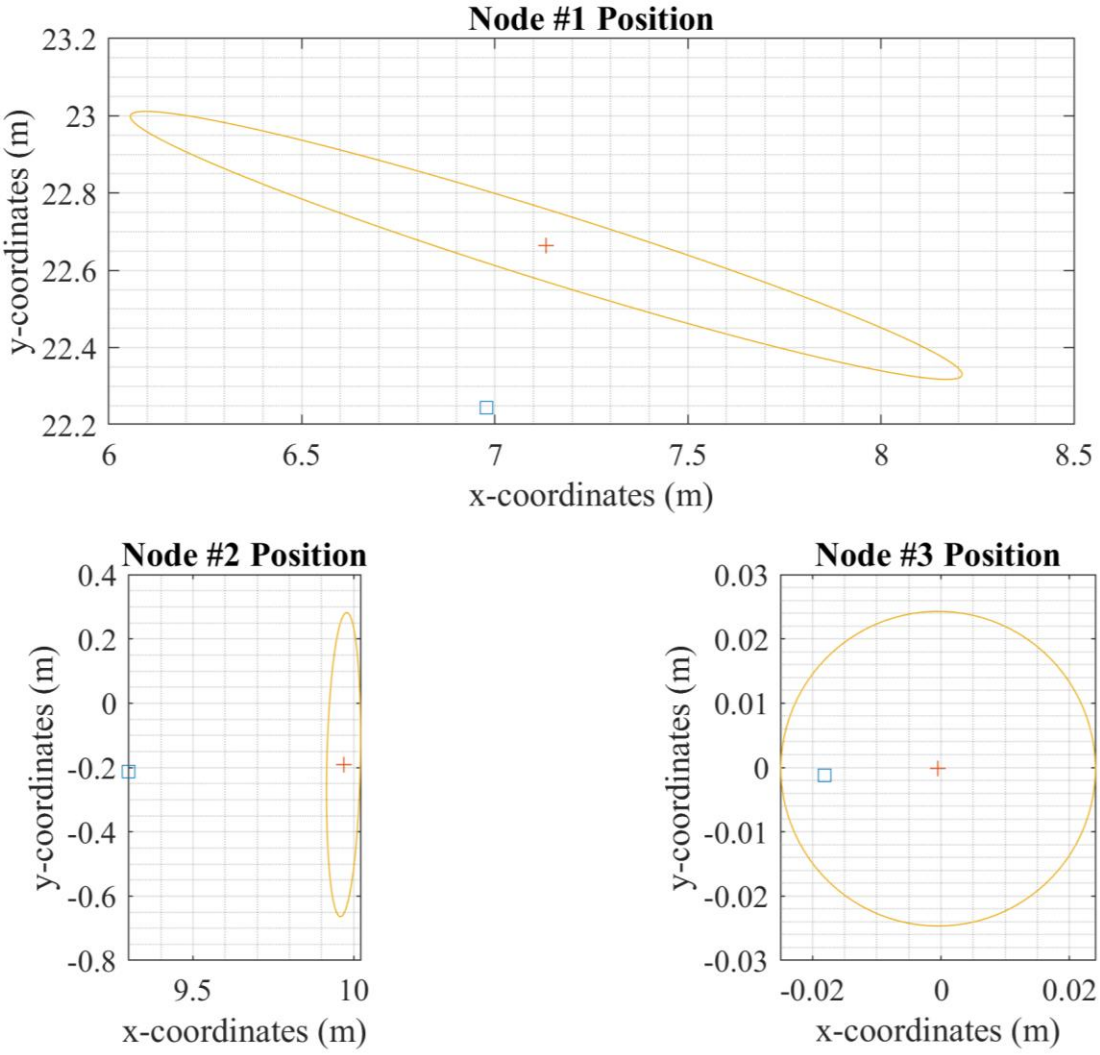


Figure 4–15 LocSpeck anchors position error ellipses

- Scenario# 7: This is the reference solution generated by the Pozyx system using the full set of Pozyx anchors.

The first two rows in Table 4–7 shows that the measurement update rate achieved using the LocSpeck system is approximately half the update rate attainable by the Pozyx system. This result is expected because of the differences between the medium access protocol implemented in the two systems. The Pozyx system supports a time-division-multiple-access protocol, while

LocSpeck depends on a random-access protocol, which intrinsically lowers the throughput of the system.

Figure 4–16 and Figure 4–17 show the estimated position of the dynamic node and the reference trajectory generated by the Pozyx system. Figure 4–16 shows the estimated and reference positions for Scenario #1, using the LocSpeck anchors only. In this figure, there is a significant error in the estimated position after 10 minutes of the beginning of the trajectory. By including two Pozyx nodes to support the LocSpeck system, the maximum error at this point is reduced significantly, as shown in Figure 4–17.

Table 4–7 shows that using three Pozyx anchors introduces a 30% reduction in the RMS error and 15% reduction in the maximum positioning error, compared to using three LocSpeck nodes with similar geometrical configurations.

However, when using the LocSpeck anchors along with two Pozyx nodes, the maximum estimated error is reduced by 69.1%, and the RMS error is reduced by 60% to reach 1.19 m. As expected, the addition of more Pozyx anchors to the pool of active nodes reduces the positioning errors of the system. For example, when using the full set of anchors from the two systems, the RMS error reaches 0.45 m, which is equivalent to the reference performance achieved by the Pozyx system.

The last two lines in Table 4–7 shows the effect of using the dynamic-positioning algorithm on positioning errors. When using the 6 Pozyx anchors without any LocSpeck nodes, the use of the LocSpeck dynamic-positioning algorithm introduces 24% error to the RMS error while increasing the maximum error to 1.94 m from 0.8 m in the reference solution.

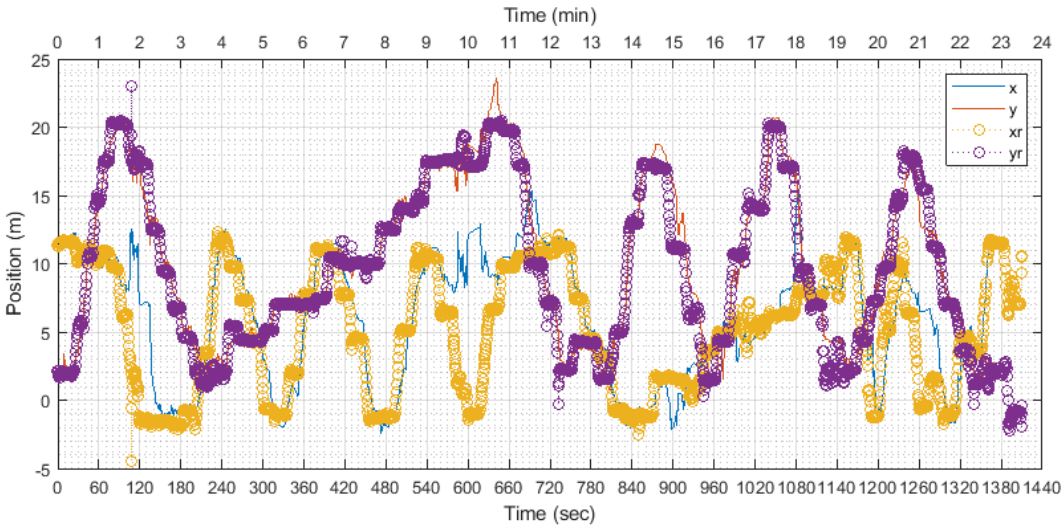


Figure 4–16 Trajectory of Scenario #1 (LocSpeck Anchors Only)

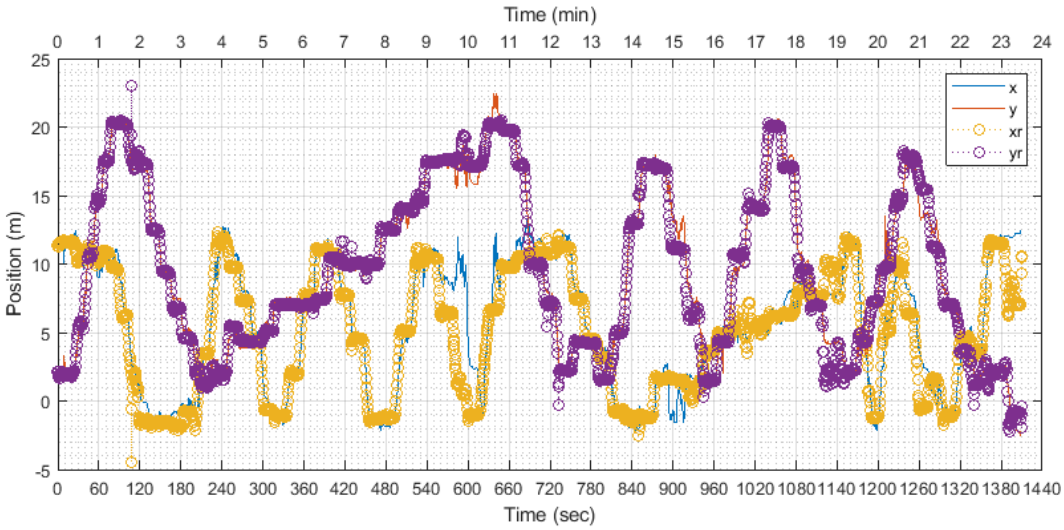


Figure 4–17 Trajectory of Scenario #3 (LocSpeck Anchors + Two Pozyx Anchors)

Table 4–7 Test Scenarios Description and Positioning Errors

#	Brief Description	Anchor #		Error 2D (m)			Average Update Rate (Hz)
		L	P	Max	Mean	RMS	
1	LOCSPECK ONLY	3	0	14	1.77	2.95	0.62
2	POZYX (6061 & 602C & 606A) - NO LOCSPECK	0	3	11.95	1.35	2.04	1.32
3	POZYX (6162 & 6056)	3	2	4.33	0.74	1.19	0.66
4	POZYX (6162 & 6056 & 6061 & 602C)	3	4	1.52	0.38	0.49	0.83
5	POZYX (6162 & 6056 & 6061 & 602C & 606A & 682d)	3	6	1.38	0.37	0.45	0.97
6	POZYX (6162 & 6056 & 6061 & 602C & 606A & 682d)	0	6	1.94	0.43	0.52	1.16
7	POZYX - Reference	0	6	0.8	0.39	0.42	1.63

4.5 Summary

This chapter presented LocSpeck, an ad-hoc UWB-based positioning system that achieves comparable positioning performance to the Pozyx system, a commercial UWB positioning system, under similar operating conditions. The proposed system can provide much more flexibility, thanks to the random medium access protocol. LocSpeck network can accommodate an arbitrary number of nodes operating in the same region. Additionally, it supports the dynamic inclusion or removal of nodes from the network.

This chapter defined the collaboration type between different nodes using relative range measurements. The UWB ranging messages are also used for node-to-node communication, which enables the sharing of states or measurements among the collaborating nodes.

Chapter Five: **Distributed Particle Filters for Collaborative Positioning**

This chapter presents the derivation of two algorithms for collaborative particle filters for the distributed estimation of nodes position using relative range measurements. The main objective of these filters is to estimate the posterior distribution of the local node position state while taking into consideration the correlation in the local states with those of the neighbouring nodes, arising from previous collaborations. This work adopts an approach similar to the Schmidt-Kalman filter [70], [176] to consider the effect of the states of the neighbouring nodes, without estimating them. This filter is referred to as the 'consider' Kalman filter [177].

The rest of this chapter is organized as follows. The chapter begins with a brief overview of recent solutions to the distributed estimation problem. Next, the proposed collaborative particle filter family is introduced. Two different variations of the collaborative particle filter are presented in details. Then, the simulation environment used to evaluate the performance of the filter is described, along with the details of the different evaluation criteria. The results are presented in the same section. Finally, a summary section concludes this chapter.

5.1 Background

This chapter addresses the collaborative position estimation problem for mobile nodes in two-dimensional global–i.e. shared among all nodes–coordinates using relative and absolute range measurements, under communication and sensing constraints. The proposed filter family was developed taking into consideration a dynamic and ad-hoc communication topology [178], in which the network topology is not constant, and the number of collaborating nodes is not constant or known in advance. Specifically, the filter is designed to estimate the marginalized posterior distribution of the position of each node, conditioned on a sequence of range measurements from fixed anchors and the neighbouring mobile nodes.

This work was initially inspired by [70], [176], in which a Schmidt-Kalman filter is used to account for the uncertainty in the states of collaborating nodes along with the correlation between the states of the collaborating nodes, arising from the relative range measurement update step. The concept of using a 'consider' filter for localization and mapping for indoor environments was evaluated in [179]. The use of particle filter in this context was proved to provide better performance for local nodes state estimation [180], especially when considering fusing information from sources such as Wi-Fi RSSI fingerprints or floorplan information [181]. The problem of distributed state estimation using relative and absolute measurements has attracted a lot of research effort in the last years. One appealing approach to solve this problem is by using the distributed particle filter approach, especially when the system or the measurement models are non-linear or described by non-Gaussian distributions. Recent surveys of the subject of distributed particle filter are presented in [182], [183]. Distributed particle filters can be categorized according to the type of data communicated among the nodes into statistics dissemination filters and measurements dissemination filters. In the first category, the nodes share the processed data, e.g. representation of the likelihood function or the locally-estimated posterior distribution. In the second category, the nodes share the local measurements. Sharing different measurements from different sensors will provide a better estimation of the underlying process. For example, in [184], the authors propose using the information state innovation (of the dimension of the state vector) instead of the conventional innovation vector (of the dimension of the observation vector).

Decentralized sensor fusion algorithm with distributed particle filters was proposed in [185], in which the different nodes share a subset of the measurements. The approach uses a selective communication algorithm, in the form of a query sent by one of the nodes with a subset of the

particles approximating its local belief. The queried node will then search its local measurements database to find the most informative measurement to send back to the query initiating node, using the Kullback-Leibler divergence between the local belief of the querying node and the local belief conditioned on the new measurement. The application of a distributed particle filter algorithm using a hierarchical sensor network is described in [186], in which a class of sensors is assigned the task of sensing the environment, and another class of sensors with more energy and computational power is responsible for handling the entire computational load. The distributed resampling with non-proportional allocation (DRNA) algorithm is implemented to parallelize the resampling step in a distributed particle filter [187]. In which a particle exchange step is implemented, where a subset of the particles is transmitted along with its unnormalized weights. The particle exchange step is followed by a local weight update and resampling steps. The main objective of this algorithm is to solve the problem of particle degeneracy resulting from the local update of the importance weights. Cooperative self-localization and distributed target tracking algorithm, through particle-based distributed belief propagation algorithm and consensus scheme, is proposed in [188]. More recent distributed particle filter algorithms are presented in [189]. Robust distributed particle filter algorithm is described in [190]. The belief propagation algorithm was extended to non-parametric belief representations [191], [192], where the belief is represented using particles. This class of algorithms generalize the sequential Monte Carlo methods to arbitrary graphs, in contrast to the temporal nature of filtering problems.

The distributed estimation problem is formulated as a set of linear equation constraints on the local coordinates of the sensor nodes with respect to the global coordinates by using local bearing information in [193]. An estimator based on the least-squares criterion for target localization using distances and angle of arrival is proposed in [194]. A different approach to

avoid the non-linear nature of the problem is proposed in [195], in which the coordinates of the nodes are reparametrized, and only a selected set of anchors are used for the localization of certain node. In [196], the problem is solved using an asynchronous consensus-based algorithm. The problem is formulated as a distributed least-squares problem. A similar formulation was discussed earlier in [197] and [198], where the optimal state estimate for each node is calculated iteratively using the Jacobi algorithm. A similar problem is presented in [199]; however, relative and absolute measurements are assumed to be available to the nodes. The problem of robust estimation from relative and unreliable measurements is addressed in [200]. In this work, the authors solve the problem using both centralized and distributed algorithms based on Least-Squares regression and Expectation Maximization algorithm (LS-EM). However, the distributed algorithm requires knowledge of the noise parameters, which is modelled as a Gaussian mixture. A different approach to solving this problem is proposed in [201], using variations on the Randomized Kaczmarz method, which is an iterative algorithm to solve linear systems of equations [202]. Other solutions are described in [203]–[205].

5.2 A Family of Distributed Particle Filters

This section describes the implementation and the derivation of the two distributed particle filters. The development of the filter family starts by the direct implementation of the collaborative particle filter, in which the considered states–the states of the remote nodes–are assumed to be linearly dependent on each particle and can be marginalized using the Rao-Blackwellized approach. The second filter makes use of a Gaussian regularization kernel for the posterior distribution, where the parameters of the regularization kernel are estimated jointly with the considered state variance and covariance using a Schmidt-Kalman filtering step.

5.2.1 Rao-Blackwellized collaborative particle filter

The formulation of the Rao-Blackwellized collaborative particle filter (RBCPF) starts by defining the augmented state vector, $\boldsymbol{x}_k$, which comprises the considered state vector, x_k^c, and the local state vector, x_k. This notation emphasizes that the considered state estimate is not important to the local estimator; only its effect on the local state is important. The posterior distribution of the local state is evaluated using the non-linear particle filter, while the posterior distribution of the considered state is evaluated using the linear Kalman filter. To put it formally, the objective of the Rao-Blackwellized filter, or the marginalized particle filter, is to evaluate the posterior distribution at time step k, given all the local observations available till step k, $y_{1:k}$.

$$p(x_{1:k}, x_k^c | y_{1:k}) \tag{5.1}$$

The objective of the collaborative particle filter is to evaluate the marginalized posterior distribution of the local, or the non-linear, states. The collaborative particle filter evaluates the marginalized posterior distribution of the main states.

$$p(x_{1:k}|y_{1:k}) = \int p(x_{1:k}, x_k^c|y_{1:k})\, dx_k^c \tag{5.2}$$

The development of the Rao-Blackwellized particle filter (RBPF) starts by factoring the joint distribution into a conditionally linear component and a nonlinear component:

$$p(x_{1:k}, x_k^c|y_{1:k}) = p(x_k^c|x_{1:k}, y_{1:k})p(x_{1:k}|y_{1:k}), \tag{5.3}$$

where the first component of the joint probability, $p(x_k^c|x_{1:k}, y_{1:k})$, is conditionally linear with respect to the non-linear part of the state vector. This conditionally linear distribution is evaluated using the Kalman filter. The second part of the joint probability, $p(x_{1:k}|y_{1:k})$, is the non-linear part, and it can be evaluated using the particle filter.

The development of the Rao-Blackwellized particle filter (RBPF) [206], [207] depends on a likelihood function of the form $p(y_k|x_k^n)$ to update the weights of the particles, and the dependence on the linear states is considered as a noise to the measurement. In this work, the dependence of the observations on both the linear and the non-linear states is considered in the development of the Rao-Blackwellized collaborative particle filter. With this assumption, the likelihood function is expressed as a Gaussian distribution with mean μ_k^y and variance R_k^y:

$$p(y_k|x_k, x_k^c) \sim \mathcal{N}\left(y_k; \mu_k^y, R_k^y\right), \tag{5.4}$$

where y_k is the range measured between two nodes, μ_k^y is the expectation of range, and R_k^y is the variance-covariance matrix of the observation. The expectation of the likelihood distribution is a function of the non-linear and linear parts of the state vector, and it is the Euclidian distance between the two nodes in two-dimensional space, and it is expressed as the 2-norm of the difference between the local and the collaborating node state:

$$\begin{aligned} \mu_k^y &= h(x_k, x_k^c) \\ &= \|x_k - x_k^c\|_2 \end{aligned} \tag{5.5}$$

Using regular particle filter to predict the joint posterior of the local and remote nodes states will require a large number of particles, which will increase exponentially with every collaboration [208]. An alternative to this approach is to use the Rao-Blackwellized PF and model the remote nodes states as linear states and perform the linear state prediction using a Kalman filter.

The non-linear component posterior can be calculated by marginalizing Equation 5.3, to integrate out the linear component of the state:

$$\begin{aligned} p(x_{1:k}|y_{1:k}) &= \int p(x_{1:k}, x_k^c|y_{1:k})\, dx_k^c \\ &= \int \frac{p(y_k|x_{1:k}, x_k^c, y_{1:k-1}) p(x_{1:k}, x_k^c|y_{1:k-1})}{p(y_k|y_{1:k-1})}\, dx_k^c \end{aligned} \tag{5.6}$$

The joint probability, $p(x_{1:k}, x_k^c|y_{1:k})$, is expanded using the regular particle filter measurement recursion equation. The term $p(y_k|y_{1:k-1})$ is a constant that does not depend on x_k^c, and can be taken out of the integration.

The posterior can be simplified as follows:

$$\begin{aligned} p(x_{1:k}|y_{1:k}) &\propto \int p(y_k|x_k, x_k^c)p(x_k^c|x_{1:k}, y_{1:k-1})p(x_{1:k}|y_{1:k-1})\, dx_k^c \\ &= \int p(y_k|x_k, x_k^c)p(x_k^c|x_{1:k}, y_{1:k-1})p(x_k|x_{1:k-1}, y_{1:k-1})p(x_{1:k-1}|y_{1:k-1})\, dx_k^c \\ &= p(x_k|x_{k-1})p(x_{1:k-1}|y_{1:k-1}) \int p(y_k|x_k, x_k^c)p(x_k^c|x_{1:k}, y_{1:k-1})\, dx_k^c \end{aligned} \tag{5.7}$$

The main difference between Equation 5.7 and a similar expression derived in [132] is the inclusion of the measurement likelihood function inside the marginalization integral since the measurement is a function of the linear and non-linear states components.

The next step in the derivation is evaluating the recursive weights equation, starting from the usual weight equation:

$$w_k = \frac{p(x_{1:k}|y_{1:k})}{q(x_{1:k}|y_{1:k})}, \tag{5.8}$$

where $q(x_{1:k}|y_{1:k})$ is the importance distribution, which can be expressed in a recursive form as follows [209]:

$$q(x_{1:k}|y_{1:k}) = q(x_k|x_{k-1})q(x_{1:k-1}|y_{1:k-1}) \tag{5.9}$$

By substituting the posterior expression from Equation 5.7, the weight update equation can be expressed as follows:

$$w_k \propto \frac{p(x_k|x_{k-1})p(x_{1:k-1}|y_{1:k-1}) \int p(y_k|x_k, x_k^c)p(x_k^c|x_{1:k}, y_{1:k-1})\, dx_k^c}{q(x_k|x_{k-1})q(x_{1:k-1}|y_{1:k-1})} \tag{5.10}$$

Noting that:

$$w_{k-1} = \frac{p(x_{1:k-1}|y_{1:k-1})}{q(x_{1:k-1}|y_{1:k-1})} \tag{5.11}$$

Following Equations 5.10 and 5.11, the recursive expression for the weights is given as:

$$w_k \propto w_{k-1} \frac{p(x_k|x_{k-1}) \int p(y_k|x_k, x_k^c) p(x_k^c|x_{1:k}, y_{1:k-1})\, dx_k^c}{q(x_k|x_{k-1})} \tag{5.12}$$

The importance distribution could take any form, depending on the exact application. However, for simplicity, let $q(x_k|x_{1:k-1}) = p(x_k|x_{k-1})$, which result in the following recursive weights expression:

$$w_k \propto w_{k-1} \int p(y_k|x_k, x_k^c) p(x_k^c|x_{1:k}, y_{1:k-1})\, dx_k^c \tag{5.13}$$

For each particle, this could be expressed as:

$$w_k^i \propto w_{k-1}^i \int p\left(y_k|x_k^i, x_k^{c,i}\right) p\left(x_k^{c,i}|x_{1:k}^i, y_{1:k-1}\right) dx_k^{c,i}, \tag{5.14}$$

where the superscript i refers to the particle number, $i = 1, \ldots, N$, and N is the total number of particles. Recall that the conditional distribution of the considered state is evaluated using the Kalman filter and takes the following form:

$$p\left(x_k^{c,i}|x_{1:k}^i, y_{1:k-1}\right) \sim \mathcal{N}\left(x_k^{c,i}; \bar{x}_k^{c,i}, \bar{P}_k^{c,i}\right) \tag{5.15}$$

where, $\bar{x}_k^{c,i}, \bar{P}_k^{c,i}$ are the estimate and the covariance matrix of the consider state, obtained from the Kalman filter update step. The measurement likelihood function can be expressed as follows:

$$p\left(y_k|x_k^i, x_k^{c,i}\right) \sim \mathcal{N}\left(y_k; \mu_k^{y,i} = h\left(x_k^i, x_k^{c,i}\right), R_k^y\right) \tag{5.16}$$

The integral in Equation 5.14 can be expressed as follows:

$$
\begin{aligned}
I &= p\left(y_k \middle| x_k^i, x_k^{c,i}\right) \times p\left(x_k^{c,i} \middle| x_{1:k}^i, y_{1:k-1}\right) \\
&= \mathcal{N}\left(y_k; h\left(x_k^i, x_k^{c,i}\right), R\right) \times \mathcal{N}\left(x_k^{c,i}; \bar{x}_k^{c,i}, \bar{P}_k^{c,i}\right) \\
&= \left(\det(2\pi R)\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\left(y_k - h_k^i\right)^T \left(R_k^y\right)^{-1}\left(y_k - h_k^i\right)\right)\right) \times \\
&\quad \left(\det\left(2\pi \bar{P}_k^{c,i}\right)\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\left(x_k^{c,i} - \bar{x}_k^{c,i}\right)^T \left(\bar{P}_k^{c,i}\right)^{-1}\left(x_k^{c,i} - \bar{x}_k^{c,i}\right)\right)\right)
\end{aligned} \tag{5.17}
$$

■

The integral given by Equation 5.17 is hard to evaluate analytically and needs to be computed numerically for every particle to update the weight associated with each particle. From the practical perspective, the limits of this integral could be set to a reasonable value to reduce the computational load for evaluating the likelihood function. Otherwise, the time required will not fit within the epoch processing time, for real-time or near real-time applications.

The derivation of the filter equations so far does not provide a mean of preserving the covariance between the current estimate of the local states and the corresponding estimates of the remote states. In the context of reduced-order Kalman filter–the Schmidt-Kalman filter–the covariance information is stored in the off-diagonal blocks of the state covariance matrix. In the case of a centralized particle filter, the joint distribution of the states of different nodes is represented by the augmented state vector and approximated as a set of particles. However, in the proposed collaborative filter, the state of the remote nodes is replaced and reset each time a new relative range measurement is received. The solution adopted for the filter is to store the covariance between the posterior distribution of the local state and the posterior of the remote node calculated locally. This covariance is used to evaluate the remote state conditional on the local state, as shown in equations shortly after. Although the joint probability is not Gaussian and needs more than the second and first moments to describe its behaviour fully, only the second-

order moment will be considered to minimize the amount of memory required to store this term. Additionally, this representation is compatible with the remote state approximation adopted for the collaborative filter. The cross-covariance of the consider and the local state is, by definition, expressed as follows:

$$\begin{aligned}\mathbf{cov}(x_k, x_k^c) &= \mathbb{E}[(x_k - \hat{x}_k)(x_k^c - \hat{x}_k^c)^T] \\ &= \mathbb{E}[x_k (x_k^c)^T] - \hat{x}_k (\hat{x}_k^c)^T,\end{aligned} \tag{5.18}$$

where $\hat{x}_k$ is the mean of the non-linear state, and $\hat{x}_k^c$ is the mean of the considered state.

The mean of the non-linear state can be taken as the weighted mean of the particles.

$$\hat{x}_k = \sum_{i=1}^{N} w_k^i x_k^i \tag{5.19}$$

The covariance of the non-linear state is expressed as follows:

$$\mathbf{cov}(x_k, x_k) = \left(\sum_{i=1}^{N} w_k^i x_k^i \left(x_k^i\right)^T \right) - \hat{x}_k (\hat{x}_k)^T \tag{5.20}$$

The posterior of the considered state is represented as a Gaussian mixture distribution of the following form:

$$p(x_k^c) = \sum_{i=1}^{N} w_k^i \mathcal{N}\left(x_k^c; x_k^{c,i}, P_k^{c,i}\right) \tag{5.21}$$

The mean of the Gaussian mixture is the weighted sum of the means of every single component, given by:

$$\begin{aligned}\hat{x}_k^c &= \sum_{i=1}^{N} w_k^i \mathbb{E}\left[\mathcal{N}\left(x_k^c; x_k^{c,i}, P_k^{c,i}\right)\right] \\ &= \sum_{i=1}^{N} w_k^i x_k^{c,i}\end{aligned} \tag{5.22}$$

The joint distribution of the consider and the local state can be expressed as follows:

$$p(x_k, x_k^c) = \sum_{i=1}^{N} w_k^i \delta\left(x_k - x_k^i\right) \mathcal{N}\left(x_k^c; x_k^{c,i}, P_k^{c,i}\right) \tag{5.23}$$

The first term in Equation 5.18 can be calculated as follows:

$$\begin{aligned}
\mathbb{E}[x_k (x_k^c)^T] &= \int \int (x_k (x_k^c)^T) p(x_k, x_k^c)\, dx_k\, dx_k^c \\
&= \int \int (x_k (x_k^c)^T) \sum_{i=1}^{N} w_k^i \delta\left(x_k - x_k^i\right) \mathcal{N}\left(x_k^c; x_k^{c,i}, P_k^{c,i}\right) dx_k\, dx_k^c \\
&= \sum_{i=1}^{N} w_k^i x_k^i \left(x_k^{c,i}\right)^T
\end{aligned} \tag{5.24}$$

Substituting Equation 5.19, 5.22, 5.24 in Equation 5.18 results in the cross-covariance matrix expression:

$$\mathbf{cov}(x_k, x_k^c) = \left(\sum_{i=1}^{N} w_k^i x_k^i \left(x_k^{c,i}\right)^T \right) - \hat{x}_k (\hat{x}_k^c)^T \tag{5.25}$$

Initially, the $\mathbf{cov}(x_k, x_k^c)$ is equal to 0; however, after the first relative update from a certain node, the cross-covariance is calculated, and the value is used afterwards in calculating the conditional distribution of the remote node. The conditional distribution equation for Gaussian distribution can be represented as follows:

$$p(x_A | x_B) \sim \mathcal{N}(\mu_A + \Sigma_{AB} \Sigma_{BB}^{-1} (x_B - \mu_B), \Sigma_{AA} - \Sigma_{AB} \Sigma_{BB}^{-1} \Sigma_{BA}) \tag{5.26}$$

The conditional distribution, $p(x_k^c | x_{1:k}, y_{1:k-1})$, is calculated as follows:

$$p(x_k^c | x_{1:k}, y_{1:k-1}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}\left(x_k^c; \bar{x}_k^{c,i}, \bar{P}_k^c\right) \tag{5.27}$$

$$\bar{x}_k^{c,i} = x_k^c + \mathbf{cov}(x_{k-1}, x_{k-1}^c) \left(\mathbf{cov}(x_{k-1}, x_{k-1})\right)^{-1} \left(x_k^i - \hat{x}_{k-1}^i\right) \tag{5.28}$$

$$\bar{P}_k^c = P_k^c - \mathbf{cov}(x_{k-1}, x_{k-1}^c) \left(\mathbf{cov}(x_{k-1}, x_{k-1})\right)^{-1} \left(\mathbf{cov}(x_{k-1}, x_{k-1}^c)\right)^T \tag{5.29}$$

where Σ_{AB} is the cross-covariance, $\mathbf{cov}(x_{k-1}, x_{k-1}^c)$, Σ_{BB} is the covariance of the local state, $\mathbf{cov}(x_{k-1}, x_{x-1})$, and Σ_{AA} is the covariance matrix of the remote state, P_k^c. Note that the conditional covariance matrix is calculated only once, while the conditional mean is evaluated for each particle. Algorithm 5–1 gives an overview of the Rao-Blackwellized Collaborative Particle filter:

ALGORITHM 5–1 THE RAO-BLACKWELLIZED COLLABORATIVE PARTICLE FILTER (RBCPF)

1. Initialization: For $i = 1, \dots, N$
    a. Particle Filter: Initialize each particle with the prior distribution, $x_0^i \sim p_0(x_0)$.
    b. Kalman Filter:
        i. Initialize the considered states mean and covariance, $\{\boldsymbol{x}_0^{c,i}, \boldsymbol{P}_0^{c,i}\} = \emptyset$.
        ii. Initialize the list of collaborating nodes ID, $\boldsymbol{nid}_0^c = \emptyset$.
2. Particle filter measurement update: For $k > 0$
    a. Process incoming node-to-node range measurement $\{nid_k^c, x_k^c, P_k^c, y_k^c\}$:
        i. If $nid_k^c \notin \boldsymbol{nid}_{k-1}^c$, augment the nodes ID set, $\boldsymbol{nid}_k^c \leftarrow \{\boldsymbol{nid}_{k-1}^c, nid_k^c\}$, and augment the mean and covariance of the consider states for each particle, with the incoming values,

$$\boldsymbol{x}_{k|k-1}^{c,i} \leftarrow \left[\left(\boldsymbol{x}_{k-1}^{c,i}\right)^T \quad (x_k^c)^T\right]^T$$

$$\boldsymbol{P}_{k|k-1}^{c,i} \leftarrow \begin{bmatrix} \boldsymbol{P}_{k-1}^{c,i} & \mathbf{0} \\ \mathbf{0} & P_k^c \end{bmatrix}$$

        ii. If $nid_k^c \in \boldsymbol{nid}_{k-1}^c$, replace the corresponding entries in the state vector, $\boldsymbol{x}_{k|k-1}^{c,i}$ and corresponding block in the covariance matrix, $\boldsymbol{P}_{k|k-1}^{c,i}$, with the conditional mean and covariance, $\{\bar{x}_k^c, \bar{P}_k^c\}$, using Equations 5.28, 5.29.

iii. Evaluate the likelihood function, $p_{n-n}^{i}\left(y_k^c \middle| x_k^i, x_k^{c,i}\right)$, using Equation 5.7.

b. Process any anchor-to-node range measurement $\{x^a, y_k^a\}$:

i. Evaluate the likelihood function given by,

$$p_{a-n}^{i}\left(y_k^a \middle| x_k^i\right) \sim \mathcal{N}\left(y_k^a; \left\|x^a - x_k^i\right\|, R_k^y\right)$$

c. Evaluate the total likelihood function, as follows:

$$p_k^i\left(y_k \middle| x_k^i\right) = \left(\int p_{n-n}^{i}\left(y_k^c \middle| x_k^i, x_k^{c,i}\right) p\left(x_k^{c,i} \middle| x_{1:k}^i, y_{1:k-1}\right) dx_k^{c,i}\right) \times \left(p_{a-n}^{i}\left(y_k^a \middle| x_k^i\right)\right),$$

where, $y_k = [(y_k^c)^T \quad (y_k^a)^T]^T$.

d. Evaluate the weights,

$$\widetilde{w}_k^i = p_k^i\left(y_k \middle| x_k^i\right),$$

then, normalize the weights: $w_k^i = \widetilde{w}_k^i / \sum_{j=0}^{N} \widetilde{w}_k^j$.

3. Resample *N* particles, such that:

$$\Pr\left\{x_k^i = x_k^j\right\} = w_k^j$$

4. KF measurement update: For $i = 1, \ldots, N$, update the mean and covariance matrix of the considered states for each particle,

$$\boldsymbol{x}_{k|k}^{c,i} = \boldsymbol{x}_{k|k-1}^{c,i} + K_k^i\left(y_k - h\left(x_k^i, x_k^{c,i}\right)\right)$$

$$\boldsymbol{P}_{k|k}^{c,i} = \left(I - K_k^i H_k^i\right)\boldsymbol{P}_{k|k-1}^{c,i}\left(I - K_k^i H_k^i\right)^T + K_k^i R_k^y \left(K_k^i\right)^T$$

$$K_k^i = \boldsymbol{P}_{k|k-1}^{c,i}\left(H_k^i\right)^T \left(H_k^i \boldsymbol{P}_{k|k-1}^{c,i}\left(H_k^i\right)^T + R_k^y\right)^{-1}$$

$$H_k^i = \frac{\partial}{\partial x^c} h_k(x_k, x_k^c) = \frac{\partial}{\partial x^c} \left\|x_k - x_k^c\right\|_2$$

5. PF time update: For $i = 1, \ldots, N$, update the particles using the predictive distribution,

$$x_{k+1}^i \sim p\left(x_{k+1}^i \middle| x_k^i, y_k\right)$$

6. KF time update: For $i = 1, \dots, N$, update the covariance matrix of the considered states,

$$\boldsymbol{P}_{k+1|k}^{c,i} = \Phi_{\mathrm{k}}^{c} \boldsymbol{P}_{k|k}^{c,i} \Phi_{\mathrm{k}}^{\mathrm{c}} + Q_k^c$$

7. PF covariance update: Calculates the cross-covariance and the variance of the particle filter, according to Equation 5.20, 5.25.
8. Advance the step index, $k \leftarrow k + 1$, and repeat from step 2.

Notes on the RBCPF Algorithm:

- The boldface state and covariance matrix, $\{\boldsymbol{x}_k^{c,i}, \boldsymbol{P}_k^{c,i}\}$, refers to the augmented version maintained by the local estimator for each particle.
- The collaborating node information sent along with each new range measurement is represented as $\{x_k^c, P_k^c\}$.

5.2.2 Rao-Blackwellized Gaussian-importance collaborative particle filter

The development of the Rao-Blackwellized Gaussian-importance collaborative particle filter (RGCPF) is addressed in this section. The RGCPF is a variant of the Rao-Blackwellized collaborative particle filter which utilizes a Gaussian regularization kernel for the posterior distribution, whose parameters are evaluated using the Schmidt-Kalman filter. The Gaussian kernel is used as the importance sampling distribution, as well. The development of the mean and variance equations for the modified Schmidt-Kalman filter (SKF), along with the updated algorithm, is presented in this section. The use of Gaussian distribution to model the importance distribution has been proposed and examined by several authors, for example, in [209], [210]. The main difference between this filter and the normal Rao-Blackwellized particle filter (RBPF)

is the order of the weight computation and the particle sampling steps. This difference is highlighted in the listing of Algorithm 5–2.

The derivation of the RGCPF algorithm starts by considering the Schmidt-Kalman filter update equations. These equations are used in two different locations: first, when generating the importance density for particles sampling; and second, when updating the covariance matrix of the considered states.

The augmented state vector contains two parts: the non-linear part corresponding to the state of the local node and the considered state part corresponding to the states of the neighbouring or collaborating nodes. Note that predicting or correcting the actual value of the states of the collaborating nodes is not important since the local node does not have the required measurement update to improve the estimation of the remote state. However, the correlation between the states is of importance.

The augmented state can be represented as follows:

$$\boldsymbol{x}_k = \begin{bmatrix} x_k \\ x_k^c \end{bmatrix} \tag{5.30}$$

where, x_k is the state of the local node, and x_k^c represents the considered states. Note, that the considered state vector size changes with time with the addition and removal of collaborating nodes.

The dynamics state space model could be expressed using the following form:

$$\begin{bmatrix} x_{k+1} \\ x_{k+1}^c \end{bmatrix} = \begin{bmatrix} A_k & 0 \\ 0 & A_k^c \end{bmatrix} \begin{bmatrix} x_k \\ x_k^c \end{bmatrix} + \begin{bmatrix} w_k \\ w_k^c \end{bmatrix} \tag{5.31}$$

where, w_k and w_k^c are zero mean white Gaussian noise.

The measurement model, in general, can be expressed as:

$$y_k = h(x_k, x_k^c) \tag{5.32}$$

For the case of 2-D range measurement, this relation is given by:

$$y_k = \|x_k - x_k^c\|_2 \tag{5.33}$$

$$\boldsymbol{H}_k = \frac{\partial}{\partial \boldsymbol{x}_k} h(x_k, x_k^c) \tag{5.34}$$

$$\boldsymbol{H}_k = [H_k \quad H_k^c] \tag{5.35}$$

The measurement update equation is the regular Kalman filter update equations, give as follows:

$$\boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k-1} + \boldsymbol{K}_k(y_k - h(x_k, x_k^c)\,) \tag{5.36}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_k R_k \boldsymbol{K}_k^T \tag{5.37}$$

where,

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T\big(\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + R_k\big)^{-1} \tag{5.38}$$

The time step subscript used in the equation above, and in the following equations can be explained as follows: $k|k-1$, refers to a quantity obtained via a time update step, using the predictive model, i.e. $p(\cdot_k\,|x_{k-1}, y_{k-1})$; the $k|k$ subscript refers to quantities estimated using the current measurement update, i.e. $p(\cdot_k\,|y_k)$. The superscript n, indicating the non-linear state, is added to some quantities related to the local state, to improve readability, and obviate any ambiguities in the cross-correlation term, P_k^{nc}, as will be shown later.

Kalman gain for this system is determined by the following equations:

$$\boldsymbol{K}_k = \begin{bmatrix} K_k^n \\ K_k^c \end{bmatrix} = \boldsymbol{P}_{k|k-1}\begin{bmatrix} (H_k^n)^T \\ (H_k^c)^T \end{bmatrix}(S_k)^{-1} \tag{5.39}$$

$$\begin{aligned} S_k &= [H_k^n \quad H_k^c]\begin{bmatrix} P_{k|k-1}^n & P_{k|k-1}^{nc} \\ \left(P_{k|k-1}^{nc}\right)^T & P_{k|k-1}^c \end{bmatrix}\begin{bmatrix} (H_k^n)^T \\ (H_k^c)^T \end{bmatrix} + R_k \\ &= H_k^n P_{k|k-1}^n (H_k^n)^T + H_k^c\left(P_{k|k-1}^{nc}\right)^T (H_k^n)^T \\ &\quad + H_k^n P_{k|k-1}^{nc} (H_k^c)^T + H_k^c P_{k|k-1}^c (H_k^c)^T + R_k \end{aligned} \tag{5.40}$$

By expanding Equation 5.39, two components of the Kalman gain–one for the nonlinear and one for the consider states–can be obtained as follows:

$$\begin{bmatrix} K_k^n \\ K_k^c \end{bmatrix} = \begin{bmatrix} P_{k|k-1}^n & P_{k|k-1}^{nc} \\ \left(P_{k|k-1}^{nc}\right)^T & P_{k|k-1}^c \end{bmatrix} \begin{bmatrix} (H_k^n)^T \\ (H_k^c)^T \end{bmatrix} (S_k)^{-1}$$
$$= \begin{bmatrix} P_{k|k-1}^n (H_k^n)^T + P_{k|k-1}^{nc} (H_k^c)^T \\ \left(P_{k|k-1}^{nc}\right)^T (H_k^n)^T + P_{k|k-1}^c (H_k^c)^T \end{bmatrix} (S_k)^{-1} \tag{5.41}$$

$$K_k^n = \left(P_{k|k-1}^n (H_k^n)^T + P_{k|k-1}^{nc} (H_k^c)^T\right)(S_k)^{-1} \tag{5.42}$$

$$K_k^c = \left(\left(P_{k|k-1}^{nc}\right)^T (H_k^n)^T + P_{k|k-1}^c (H_k^c)^T\right)(S_k)^{-1} \tag{5.43}$$

The different components of the state vector and the covariance matrix in the above equations are needed in different time steps during the execution of the algorithm. Consequently, the state vector and the covariance matrix will be expressed in terms of their block components:

First, for the state vector update equation:

$$x_{k|k} = x_{k|k-1} + K_k^n \left(y_k - h\left(x_{k|k-1}, x_{k|k-1}^c\right)\right) \tag{5.44}$$

$$x_{k|k}^c = x_{k|k-1}^c + K_k^c \left(y_k - h\left(x_{k|k-1}, x_{k|k-1}^c\right)\right) \tag{5.45}$$

Second, the covariance matrix update equation:

$$\boldsymbol{P}_{k|k} = \begin{bmatrix} P_{k|k}^n & P_{k|k}^{nc} \\ \left(P_{k|k}^{nc}\right)^T & P_{k|k}^c \end{bmatrix} \tag{5.46}$$

$$\boldsymbol{P}_{k|k} = \left(\boldsymbol{I} - \begin{bmatrix} K_k^n \\ K_k^c \end{bmatrix} \begin{bmatrix} H_k^n & H_k^c \end{bmatrix}\right) \begin{bmatrix} P_{k|k-1}^n & P_{k|k-1}^{nc} \\ \left(P_{k|k-1}^{nc}\right)^T & P_{k|k-1}^c \end{bmatrix} \left(\boldsymbol{I} - \begin{bmatrix} K_k^n \\ K_k^c \end{bmatrix} \begin{bmatrix} H_k^n & H_k^c \end{bmatrix}\right)^T$$
$$+ \begin{bmatrix} K_k^n \\ K_k^c \end{bmatrix} R_k \begin{bmatrix} (K_k^n)^T & (K_k^c)^T \end{bmatrix} \tag{5.47}$$

$$P_{k|k}^{n} = (I - K_k^n H_k^n) P_{k|k-1}^{n} (I - K_k^n H_k^n)^T - (I - K_k^n H_k^n) P_{k|k-1}^{nc} (K_k^n H_k^c)^T$$
$$-K_k^n H_k^c \left(P_{k|k-1}^{nc}\right)^T (I - K_k^n H_k^n)^T + K_k^n H_k^c P_{k|k-1}^{c} (K_k^n H_k^c)^T + K_k^n R_k (K_k^n)^T \quad (5.48)$$

$$P_{k|k}^{nc} = -(I - K_k^n H_k^n) P_{k|k-1}^{n} (K_k^c H_k^n)^T + (I - K_k^n H_k^n) P_{k|k-1}^{nc} (I - K_k^c H_k^c)^T$$
$$+K_k^n H_k^c \left(P_{k|k-1}^{nc}\right)^T (K_k^c H_k^n)^T - K_k^n H_k^c P_{k|k-1}^{c} (I - K_k^c H_k^c)^T + K_k^n R_k (K_k^c)^T \quad (5.49)$$

$$P_{k|k}^{c} = K_k^c H_k^n P_{k|k-1}^{n} (K_k^c H_k^n)^T - K_k^c H_k^n P_{k|k-1}^{nc} (I - K_k^c H_k^c)^T$$
$$-(I - K_k^c H_k^c) \left(P_{k|k-1}^{nc}\right)^T (K_k^c H_k^n)^T$$
$$+(I - K_k^c H_k^c) P_{k|k-1}^{c} (I - K_k^c H_k^c)^T + K_k^c R_k (K_k^c)^T \quad (5.50)$$

Equations 5.44, 5.45, 5.48–5.50 can be simplified following the reduced-order or Schmidt-Kalman filter approach by setting the Kalman gain for the considered state, K_k^c, to 0 [129], [211]. The resulting update equations will not update the value of the consider state, while keeping track of its effect on the covariance of the local state.

The simplified version of Equations 5.48–5.50 are expressed as follows:

$$P_{k|k}^{n} = (I - K_k^n H_k^n) P_{k|k-1}^{n} - K_k^n H_k^c \left(P_{k|k-1}^{nc}\right)^T \quad (5.51)$$

$$P_{k|k}^{nc} = (I - K_k^n H_k^n) P_{k|k-1}^{nc} - K_k^n H_k^c P_{k|k-1}^{c} \quad (5.52)$$

$$P_{k|k}^{c} = P_{k|k-1}^{c} \quad (5.53)$$

The time update equations for the system defined by the state-space model in Equation 5.31 can be expressed as follows:

$$\boldsymbol{x}_{k+1|k} = \boldsymbol{\Phi}_k \boldsymbol{x}_{k|k} \quad (5.54)$$

$$\begin{bmatrix} x_{k+1|k} \\ x_{k+1|k}^c \end{bmatrix} = \begin{bmatrix} \Phi_k^n & \mathbf{0} \\ \mathbf{0} & \Phi_k^c \end{bmatrix} \begin{bmatrix} x_{k|k} \\ x_{k|k}^c \end{bmatrix} \quad (5.55)$$

$$x_{k+1|k} = \Phi_k^n x_{k|k} \quad (5.56)$$

$$x_{k+1|k}^c = \Phi_k^c x_{k|k}^c \quad (5.57)$$

The covariance matrix calculations are given by the following equations:

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{\Phi}_k \boldsymbol{P}_{k|k} (\boldsymbol{\Phi}_k)^T + \boldsymbol{Q}_k \quad (5.58)$$

$$\begin{bmatrix} P^n_{k+1|k} & P^{nc}_{k+1|k} \\ \left(P^{nc}_{k+1|k}\right)^T & P^c_{k+1|k} \end{bmatrix} = \begin{bmatrix} \Phi^n_k & \mathbf{0} \\ \mathbf{0} & \Phi^c_k \end{bmatrix} \begin{bmatrix} P^n_{k|k} & P^{nc}_{k|k} \\ \left(P^{nc}_{k|k}\right)^T & P^c_{k|k} \end{bmatrix} \begin{bmatrix} (\Phi^n_k)^T & \mathbf{0} \\ \mathbf{0} & (\Phi^c_k)^T \end{bmatrix} + \begin{bmatrix} Q^n_k & \mathbf{0} \\ \mathbf{0} & Q^c_k \end{bmatrix} \tag{5.59}$$

$$P^n_{k+1|k} = \Phi^n_k P^n_{k|k} (\Phi^n_k)^T + Q^n_k \tag{5.60}$$

$$P^{nc}_{k+1|k} = \Phi^n_k P^{nc}_{k|k} (\Phi^c_k)^T \tag{5.61}$$

$$P^c_{k+1|k} = \Phi^c_k P^c_{k|k} (\Phi^c_k)^T + Q^c_k \tag{5.62}$$

The next step is to calculate the conditional distribution of the linear state, $p(x^c_k | x_{1:k}, y_{1:k-1})$. The conditional distribution of the considered state given the non-linear state can be calculated using Equation 5.26 and using the terms calculated in Equations 5.36–5.50. This relation can be expressed as follows:

$$p\left(x^{c,i}_k \middle| x_k, y_{1:k-1}\right) \sim \mathcal{N}\left(x^{c,i}_k; \bar{x}^{c,i}_{k|k-1}, \bar{P}^{c,i}_{k|k-1}\right) \tag{5.63}$$

where,

$$\bar{x}^{c,i}_{k|k-1} = x^{c,i}_{k|k-1} + P^{nc}_{k|k-1}\left(P^n_{k|k-1}\right)^{-1}\left(x^i_k - x^i_{k|k-1}\right) \tag{5.64}$$

$$\bar{P}^{c,i}_{k|k-1} = P^c_{k|k-1} - P^{nc}_{k|k-1}\left(P^n_{k|k-1}\right)^{-1}\left(P^{nc}_{k|k-1}\right)^T \tag{5.65}$$

Note that $x^i_{k|k-1}$ is the mean of the Gaussian kernel for the i^{th} particle, while x^i_k is the value of the particle sampled from this kernel, such that $p\left(x^i_k \middle| x^i_{0:k-1}, y_{1:k}\right) \sim \mathcal{N}\left(x^i_k; x^i_{k|k}, P^{n,i}_{k|k}\right)$. In contrast to the RBCPF algorithm, the conditional covariance matrix in Equation 5.65 is updated for each particle.

$$p\left(x^{c,i}_k \middle| x_k, y_{1:k-1}\right) = \left(\det\left(2\pi \bar{P}^{c,i}_{k|k-1}\right)\right)^{-\frac{1}{2}} \times \exp\left(-\frac{1}{2}\left(\left(x^{c,i}_k - \bar{x}^{c,i}_{k|k-1}\right)\left(\bar{P}^{c,i}_{k|k-1}\right)^{-1}\left(x^{c,i}_k - \bar{x}^{c,i}_{k|k-1}\right)^T\right)\right) \tag{5.66}$$

The integral of the marginal likelihood function can be modified according to Equation 5.67:

$$
\begin{aligned}
I &= p\left(y_k \middle| x_k^i, x_k^{c,i}\right) \times p\left(x_k^{c,i} \middle| x_{1:k}^i, y_{1:k-1}\right) \\
&= \mathcal{N}\left(y_k; h\left(x_k^i, x_k^{c,i}\right), R\right) \times \mathcal{N}\left(x_k^{c,i}; \bar{x}_{k|k}^{c,i}, \bar{P}_{k|k}^{c,i}\right) \\
&= \left(\det(2\pi R)\right)^{-\frac{1}{2}} \\
&\quad \times \exp\left(-\frac{1}{2}\left(\left(y_k - h_k^i\right)^T \left(R_k^y\right)^{-1}\left(y_k - h_k^i\right)\right)\right) \\
&\quad \times \left(\det\left(2\pi \bar{P}_{k|k}^{c,i}\right)\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\left(x_k^{c,i} - \bar{x}_{k|k}^{c,i}\right)^T \left(\bar{P}_{k|k}^{c,i}\right)^{-1}\left(x_k^{c,i} - \bar{x}_{k|k}^{c,i}\right)\right)\right)
\end{aligned} \tag{5.67}
$$

■

The RGCPF algorithm can be summarized in the Algorithm 5–2 listing:

ALGORITHM 5–2 THE RAO-BLACKWELLIZED GAUSSIAN POSTERIOR COLLABORATIVE PARTICLE FILTER (RGCPF)

1. Initialization: For $i = 1, \ldots, N$

    a. Particle Filter:

    Initialize each particle using the prior distribution, $x_0^i \sim p_0(x_0)$.

    b. Kalman Filter:

    Initialize the state vector and covariance matrix, $\left\{\boldsymbol{x}_{0|0}^i, \boldsymbol{P}_{0|0}^i\right\}$, such that:

        i. Initialize the non-linear states and covariance, $\left\{x_{0|0}^i, P_{0|0}^{n,i}\right\}$.

        ii. Initialize the consider states and covariance, $\left\{x_{0|0}^{c,i}, P_{0|0}^{c,i}\right\} = \emptyset$.

        iii. Initialize the list of collaborating nodes ID, $\boldsymbol{nid}_0^c = \emptyset$.

2. Process incoming node-to-node range measurement $\{nid_k^c, x_k^c, P_k^c, y_k^c\}$:

    If $nid_k^c \notin \boldsymbol{nid}_{k-1}^c$, augment the nodes ID set, $\boldsymbol{nid}_k^c \leftarrow \{\boldsymbol{nid}_{k-1}^c, nid_k^c\}$, and augment the state vector and the covariance matrix of the consider states for each particle, with the incoming values,

$$
\begin{aligned}
\boldsymbol{x}_{k|k-1}^i &\leftarrow \begin{bmatrix} \left(\boldsymbol{x}_{k|k-1}^i\right)^T & \left(x_{k|k-1}^c\right)^T \end{bmatrix}^T \\
\boldsymbol{P}_{k|k-1}^i &\leftarrow \begin{bmatrix} \boldsymbol{P}_{k|k-1}^i & \boldsymbol{0} \\ \boldsymbol{0} & P_{k|k-1}^c \end{bmatrix}
\end{aligned}
$$

    Otherwise, update the corresponding entries in the state vector and the covariance matrix.

3. SKF measurement update for the non-linear part of the state vector: For $i = 1, \dots, N$, update the state and the covariance matrix of the particles, to get $\{x_{k|k}^i, P_{k|k}^{n,i}\}$, using Equation 5.44, 5.48.

4. Sample from the importance distribution, given by:

$$x_k^i \sim q(x_k | x_{0:k-1}^i, y_{0:k})$$

where,

$$q(x_k | x_{0:k-1}^i, y_{1:k}) = \mathcal{N}(x_k; x_{k|k}^i, P_{k|k}^{n,i})$$

5. Particle filter measurement update:

   a. Evaluate the total likelihood function, as follows:

$$p_k^i(y_k | x_k^i) = \int p_{n-n}^i(y_k^c | x_k^i, x_k^{c,i}) p(x_k^{c,i} | x_{1:k}^i, y_{1:k-1})\, dx_k^{c,i} \times p_{a-n}^i(y_k^a | x_k^i)$$

   where, $y_k = [(y_k^c)^T \quad (y_k^a)^T]^T$.

   b. Evaluate the weights,

$$w_k^i = w_{k-1}^i \frac{p(y_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, y_{1:k})}$$

   then, normalize the weights: $\widetilde{w}_k^i = w_k^i / \sum_{j=0}^N w_k^j$.

6. If needed, resample N particles, such that:

$$p(x_k^i = x_k^j) = \widetilde{w}_k^j$$

7. KF measurement update: Update the state and the covariance matrix of the consider states to get $\{x_{k|k}^{c,i}, P_{k|k}^{nc,i}, P_{k|k}^{c,i}\}$, according to Equation 5.45, 5.49, 5.50.

8. PF time update: For $i = 1, \dots, N$, update the particles according to the predictive distribution,

$$x_{k+1}^i \sim p(x_{k+1}^i | x_k^i, y_k)$$

9. KF time update: For $i = 1, \dots, N$, update the covariance matrix of the consider states,

$$\boldsymbol{P}_{k+1}^{c,i} = \Phi_{\mathrm{k}}^c \boldsymbol{P}_k^{c,i} \Phi_{\mathrm{k}}^{\mathrm{c}} + Q_k^c$$

10. Advance the step index, $k \leftarrow k + 1$, and repeat from step 2.

5.3 Simulation and results

The performance of the proposed particle filter family is evaluated through simulation against a centralized and distributed particle filters. In the centralized filter, the states of all nodes are concatenated into one state vector. Thus, the joint distribution of the state of the entire node population is evaluated with each measurement update. The main challenge to this approach is the particle depletion problem, since the dimensionality of the centralized filter in M times that of the distributed filter, where M is the total number of nodes.

The second filter configuration is the distributed filter configuration. In this case, each node estimates its state locally, according to range measurement updates either to the reference anchors or to the neighbouring nodes. In this configuration, the anchors and neighbouring nodes are treated equally, such that the uncertainty in the remote state estimates will be ignored by the local particle filter during the evaluation of the measurement likelihood function. Additionally, the local particle filter in this configuration is oblivious to the correlation between the local states and the remote node states.

The performance is evaluated using the root mean square error (RMSE) of the positioning derived from the Cramér-Rao Lower Bound (CRLB) of each estimator. The CRLB criterion is useful to evaluate how the filter is behaving against the most efficient theoretical estimator of the state given the available measurements. The following section will provide a brief background of the CRLB along with the computation of the bound for the intended measurement and dynamical models.

5.3.1 Cramér-Rao Lower Bound

Cramér-Rao Lower Bound (CRLB) defines a lower bound for the variance of any unbiased estimator. For any unbiased estimator, $\hat{x}_{k|k}$, using a series of measurements, $y_{1:k}$, with a likelihood of $p(y_{1:k}|x_{0:k})$, given an initial a prior distribution, $p(x_0)$, the variance is lower-bounded using the following equation:

$$\text{cov}\big(\hat{x}_{k|k}\big) \geq \mathbf{J}_F^{-1} \tag{5.68}$$

where $\mathbf{J}_F$ is the Fisher information matrix (FIM) [212], [213].

The Fisher information matrix, $\mathbf{J}_F$, is a $n_x \times n_x$ matrix, where the element (i, j) of this matrix can be expressed using the following relation:

$$[\mathbf{J}_F]_{(i,j)} \triangleq -\mathbb{E}\left\{\frac{\partial^2 \ln\big(p(y,x)\big)}{\partial x_{(i)} \partial x_{(j)}}\right\} \tag{5.69}$$

Alternatively,

$$\mathbf{J}_F \triangleq -\mathbb{E}\{\nabla_x [\nabla_x \ln p(y,x)]^T\} \tag{5.70}$$

■

The expression in Equation 5.70 can be evaluated recursively for estimating a state vector $x_{1:k}$ as follows [212], [214], for $k > 0$:

$$(\mathbf{J}_F)_{k+1} = D_k^{22} - D_k^{21}[(\mathbf{J}_F)_k + D_k^{11}]^{-1} D_k^{12} \tag{5.71}$$

where,

$$D_k^{11} = -\mathbb{E}\{\nabla_{x_k}\big[\nabla_{x_k} \ln p(x_{k+1}|x_k)\big]^T\} \tag{5.72}$$

$$D_k^{21} = -\mathbb{E}\{\nabla_{x_k}\big[\nabla_{x_{k+1}} \ln p(x_{k+1}|x_k)\big]^T\} \tag{5.73}$$

$$D_k^{12} = -\mathbb{E}\{\nabla_{x_{k+1}}\big[\nabla_{x_k} \ln p(x_{k+1}|x_k)\big]^T\} = [D_k^{12}]^T \tag{5.74}$$

$$D_k^{22} = -\mathbb{E}\{\nabla_{x_{k+1}}\big[\nabla_{x_{k+1}} \ln p(x_{k+1}|x_k)\big]^T\} - \mathbb{E}\{\nabla_{x_{k+1}}\big[\nabla_{x_{k+1}} \ln p(y_{k+1}|x_{k+1})\big]^T\} \tag{5.75}$$

The initial Fisher information matrix at $k = 0$ depends in the initial a priori distribution, $p(x_0)$, and is given as follows:

$$(\mathbf{J}_F)_0 = -\mathbb{E}\{\nabla_{x_0}\left[\nabla_{x_0} \ln p(x_0)\right]^T\} \tag{5.76}$$

■

Discussion about the Cramér-Rao Lower Bound for the distributed positioning architecture can be found in [215], which provides an example for a set of collaborating nodes in a small region. All the nodes are stationary and share measurements through anchors and other nearby nodes. Other resources covering the same topic is given in [212], [216]–[220].

The bound provided by the previous equations above is called the joint unconditional CRLB, and it does not provide the tightest bound [221], meaning that the actual performance of the optimal estimator may fall far from that lower bound. However, for this work, the real value of the bound is not significant as it is just an indication of the achievable error variance.

The Cramér-Rao lower bound of the filter is calculated for the case of the intermittent observations. In which only one active connection is used to achieve the relative range measurement. As opposed to a continues observation mode, where each node can obtain multiple range measurements simultaneously. The rest of this section is used to derive the CRLB to be used to evaluate the RMSE performance of the filters. The bound adopted for the evaluation of the performance of the filter is the joint unconditional bound.

The first step is to evaluate the initial information matrix $(\mathbf{J}_F)_0$. Assuming a Gaussian prior distribution, the initial two-dimension position is described as a group of uncorrelated nodes with a certain mean that corresponds to the actual location and a variance.

$$p(x_0) = \mathcal{N}(x_0; \mu_0, \Sigma_0) \tag{5.77}$$

where $\mu_0 = [(x_1)^T \quad ... \quad (x_n)^T]^T$, $\Sigma_0 = \sigma_0^2 \boldsymbol{I}$, $\boldsymbol{I}$ is the $2n \times 2n$ identity matrix, and n is the total number of nodes.

$$p(x_0) = |2\pi\Sigma_0|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0)\right) \tag{5.78}$$

$$\log p(x_0) = -\frac{1}{2}\log|2\pi\Sigma_0| - \frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0) \tag{5.79}$$

The expectation of the log-likelihood is calculated with respect to the prior distribution, $p(x_0)$.

$$\nabla_{x_0} \ln p(x_0) = \nabla_{x_0}\left(-\frac{1}{2}\log|2\pi\Sigma_0| - \frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0)\right) \tag{5.80}$$

$$= -\Sigma_0^{-1}(x_0 - \mu_0) \tag{5.81}$$

$$\begin{aligned} \nabla_{x_0}\left[\nabla_{x_0} \ln p(x_0)\right]^T &= \nabla_{x_0}\left(-\Sigma_0^{-1}(x_0 - \mu_0)\right)^T \\ &= \nabla_{x_0}(-(x_0 - \mu_0)^T \Sigma_0^{-1}) \\ &= -\Sigma_0^{-1} \end{aligned} \tag{5.82}$$

$$\begin{aligned} (\mathbf{J}_F)_0 &= -\mathbb{E}\{\nabla_{x_0}\left[\nabla_{x_0} \ln p(x_0)\right]^T\} \\ &= \mathbb{E}\{\Sigma_0^{-1}\} \\ &= \Sigma_0^{-1} \end{aligned} \tag{5.83}$$

Assuming a random walk dynamic model and a Gaussian measurement model, the parameters of the recursive Fisher information matrix, Equations 5.72–5.75, can be expressed as follows:

$$p(x_{k+1}|x_k) = |2\pi\Sigma_0|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_{k+1} - x_k)^T \Sigma_k^{-1}(x_{k+1} - x_k)\right) \tag{5.84}$$

$$\log p(x_{k+1}|x_k) = -\frac{1}{2}\log|2\pi\Sigma_k| - \frac{1}{2}(x_{k+1} - x_k)^T \Sigma_k^{-1}(x_{k+1} - x_k) \tag{5.85}$$

The gradient of the log-likelihood function with respect to x_k and x_{k+1} can be expressed as follows:

$$\nabla_{x_k} \log p(x_{k+1}|x_k) = \Sigma_k^{-1}(x_{k+1} - x_k) \tag{5.86}$$

$$\nabla_{x_{k+1}} \log p(x_{k+1}|x_k) = -\Sigma_k^{-1}(x_{k+1} - x_k) \tag{5.87}$$

$$\nabla_{x_k}\left[\nabla_{x_k}\log p(x_{k+1}|x_k)\right]^T = -\Sigma_k^{-1} \tag{5.88}$$

$$\begin{aligned}\nabla_{x_{k+1}}\left[\nabla_{x_k}\log p(x_{k+1}|x_k)\right]^T &= \nabla_{x_k}\left[\nabla_{x_{k+1}}\log p(x_{k+1}|x_k)\right]^T \\ &= \Sigma_k^{-1}\end{aligned} \tag{5.89}$$

$$\nabla_{x_{k+1}}\left[\nabla_{x_{k+1}}\log p(x_{k+1}|x_k)\right]^T = -\Sigma_k^{-1} \tag{5.90}$$

The measurement likelihood function is represented as:

$$\begin{aligned}p(y_{k+1}|x_{k+1}) &= |2\pi R_{k+1}|^{-\frac{1}{2}} \\ &\quad \times \exp\left(-\frac{1}{2}\left(y_{k+1}-h(x_{k+1})\right)^T \mathrm{R}_{k+1}^{-1}\left(y_{k+1}-h(x_{k+1})\right)\right)\end{aligned} \tag{5.91}$$

$$\begin{aligned}\log p(y_{k+1}|x_{k+1}) &= -\frac{1}{2}\log|2\pi R_{k+1}| \\ &\quad -\frac{1}{2}\left(y_{k+1}-h(x_{k+1})\right)^T \mathrm{R}_{k+1}^{-1}\left(y_{k+1}-h(x_{k+1})\right)\end{aligned} \tag{5.92}$$

$$\nabla_{x_{k+1}}\ln p(y_{k+1}|x_{k+1}) = -\frac{1}{2}\cdot\nabla_{x_{k+1}}\left(\left(y_{k+1}-h(x_{k+1})\right)^T \mathrm{R}_{k+1}^{-1}\left(y_{k+1}-h(x_{k+1})\right)\right) \tag{5.93}$$

$$= \left(y_{k+1}-h(x_{k+1})\right)^T \mathrm{R}_{k+1}^{-1}\left(\nabla_{x_{k+1}}h(x_{k+1})\right) \tag{5.94}$$

$$\nabla_{x_{k+1}}\left[\nabla_{x_{k+1}}\ln p(y_{k+1}|x_{k+1})\right]^T = -\left(\nabla_{x_{k+1}}h(x_{k+1})\right)^T \mathrm{R}_{k+1}^{-1}\left(\nabla_{x_{k+1}}h(x_{k+1})\right) \tag{5.95}$$

where,

$$H_{k+1} = \nabla_{x_{k+1}}h(x_{k+1}) \tag{5.96}$$

$$\begin{aligned}(H_{k+1})_i &= \frac{\partial}{\partial x_i}h(x_{k+1}) = \frac{\partial}{\partial x_i}\left\|x_i - x_j\right\|_2 \\ &= \frac{x_i - x_j}{\left\|x_i - x_j\right\|_2}\end{aligned} \tag{5.97}$$

The parameters of the recursive relation can be expressed as follows:

$$D_k^{11} = \Sigma_k^{-1} \tag{5.98}$$

$$D_k^{21} = D_k^{12} = -\Sigma_k^{-1} \tag{5.99}$$

$$D_k^{22} = \Sigma_k^{-1} + \mathbb{E}\{H_{k+1}^T R_{k+1}^{-1} H_{k+1}\} \tag{5.100}$$

The calculation of the expectation in Equation 5.100 is not straight forward. Note that the expectation is taken with respect to the state vector. One solution is to use numerous simulations and replace the expectation operator with the ensemble average of the trajectories [212], [214].

5.3.2 Results

The performance of the proposed filter family is evaluated using a two-dimensional simulation environment, running two static scenarios. The objective of both scenarios is to estimate the unknown positions of two different sets of static nodes using a combination of relative and absolute range measurements. The relative measurements are between the nodes, and the absolute measurements are between the nodes and the reference anchors. All nodes share the knowledge of the position of the anchors; however, the location estimates of the nodes are not shared until a relative measurement occurs. During a relative range measurement exchange, the two participating nodes exchange their most recent estimates in addition to the corresponding covariance. As outlined in a previous section, the state posterior of the nodes is approximated using a Gaussian distribution to reduce the communication overhead.

Figure 5–1 and Figure 5–2 show the floorplan of the simulation environment, the location of nodes and anchors, and the range of the sensors. The floorplan in both test cases is a rectangular, empty plan with four anchors at the corners of the floorplan. None of the nodes can get a position fix using absolute measurements only and need to collaborate with other nodes to do so. Figure 5–1 shows the layout of the four nodes case, which is a 40 m by 30 m rectangle. The upper-right and the lower-left nodes can measure ranges to one anchor point and can collaborate with another two nodes. The nodes on the upper-left and the lower-right corners can measure ranges between themselves, two anchors, and the two other nodes. The layout of the eight nodes case is shown in Figure 5–2, which is a 60 m by 40 m rectangle. The nodes form a symmetric lattice.

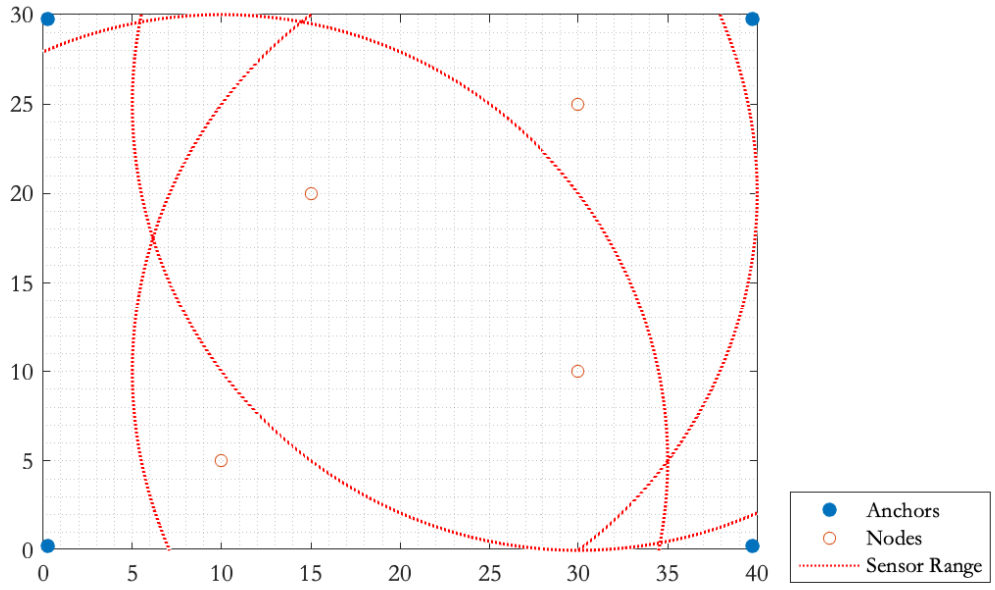


Figure 5–1 Floorplan, 4 Nodes

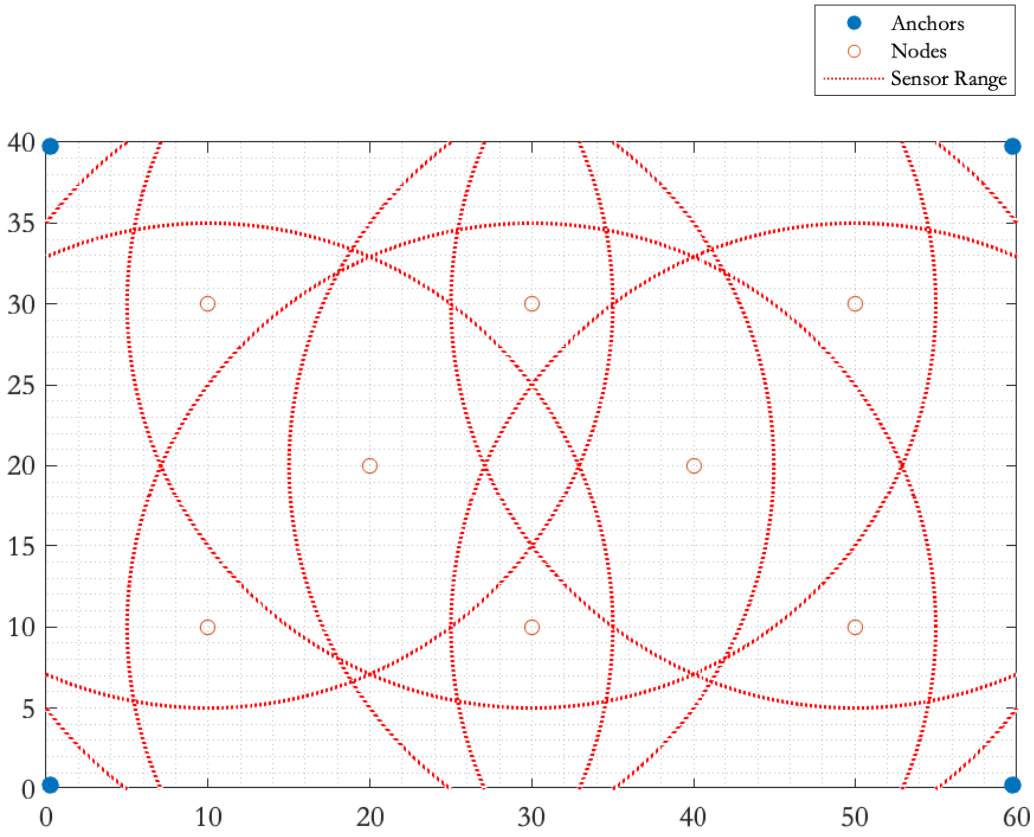


Figure 5–2 Floorplan, 8 Nodes

Each of the four corner nodes is connected to one anchor, another corner node, and two middle nodes. The middle nodes are connected to each other and to the two corner nodes. None of the middle nodes can connect directly to the anchor nodes.

The range measurements between nodes are intermittent, such that at each time step nodes can communicate with only one node or anchor. The intermittent measurement model is based on the expected behaviour of an ad-hoc network of nodes equipped with radio ranging transceivers, such as UWB-based ranging transceivers. Such ranging devices are assumed to provide unbiased distance measurement. The nodes and anchors are paired randomly, as a result of the ad-hoc nature of the sensor network topology examined in this work. The effect of the random ranging measurements structure is seen in Figure 5–3 and Figure 5–4. These figures represent the connectivity graph between all nodes by the end of the simulation time of sample run for the four nodes and the eight nodes cases, respectively. The vertices of the graph represent the nodes and anchors and are labelled accordingly. The edges represent the range measurement between two vertices, and each is weighted by the frequency of the measurement. The number on the edges is the total number of measurements between the connecting vertices. It is clear from Figure 5–3 and Figure 5–4 that this configuration poses a challenge to the positioning application since the quality of the geometry of the measurement is not guaranteed, and there are weak links in the graph, where the absence of range measurements can reduce the positioning accuracy of the nodes set.

The performance of the filter is characterized by two parameters: the average of the RMSE for 500 sample runs of the two proposed filters for 30 seconds, with 2 Hz range measurement update rate, and the effective number of particles after the weight update step for a sample run.

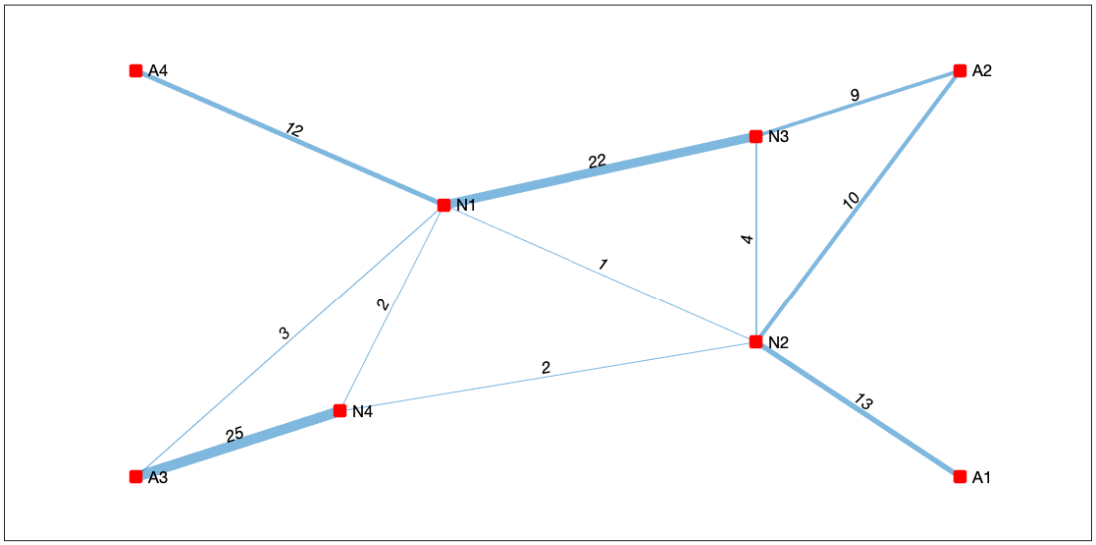


Figure 5–3 Connectivity Graph, 4 Nodes

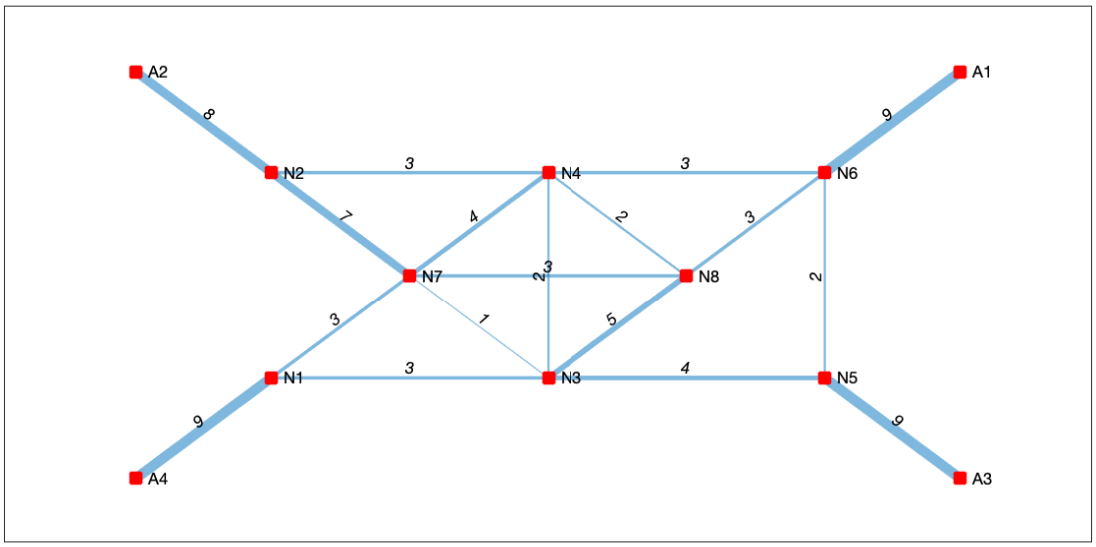


Figure 5–4 Connectivity Graph, 8 Nodes

The RMSE is compared to that of a centralized particle filter and a distributed particle filter in addition to the Cramér-Rao Lower Bound, for the two simulation scenarios–four and eight nodes. Figure 5–5 and Figure 5–6 compare the RMSE performance of the proposed filters and the two reference implementations for the two testing scenarios. The implementation of the three

distributed filters uses 150 particles for the state of each filter, while the centralized implementation utilizes 600 and 1200 particles to represent the joint distribution of the states for the four and eight cases respectively.

In these two simulation scenarios, the average RMSE performance of the RBCPF is shown to be more conforming with the CRLB than the other three filters. It is clear that the error in the case of the RBCPF is converging to the lower-bound at a higher rate. In the case of four nodes, the RGCPF is matching the performance of the RBCPF closely. However, in the case of the eight nodes, the RGCPF lags in terms of RMSE performance and is matching the performance of the distributed particle filter. In all cases, the RMSE performance of the centralized particle filter is trailing behind the other filters.

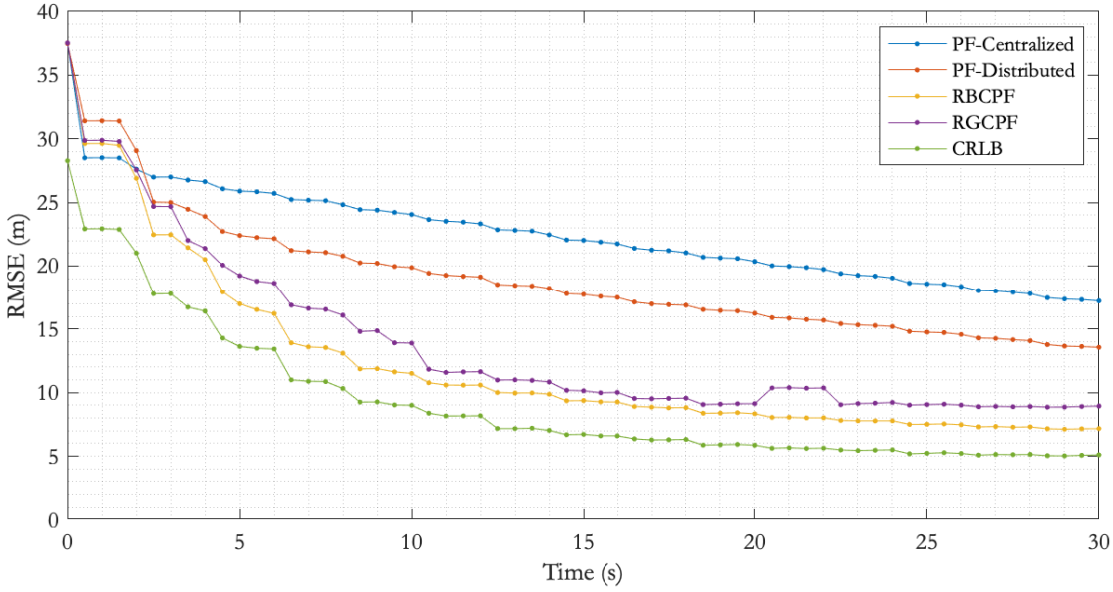


Figure 5–5 RMSE Using CRLB and Particle Filters–150 Particles, 4 Nodes, 500 Runs

The reasons why the centralized and the distributed particle filters are falling behind the RBCPF are different. The centralized particle filter suffers from the curse of the dimensionality since the number of particles required to represent the state will increase exponentially with the addition of new states. This can be confirmed in Figure 5–7 and Figure 5–8 showing the effective number of particles for the four filters. For the case of the particle filter, the percentage of the effective particles is remarkedly lower than others since only a few particles at any moment will be representing the true joint state of the nodes. On the other hand, the distributed particle filter does not consider the correlation between the nodes, so it is faster to converge to a solution, as shown in Figure 5–7 and Figure 5–8, albeit not necessarily the correct solution as evident in Figure 5–5 and Figure 5–6.

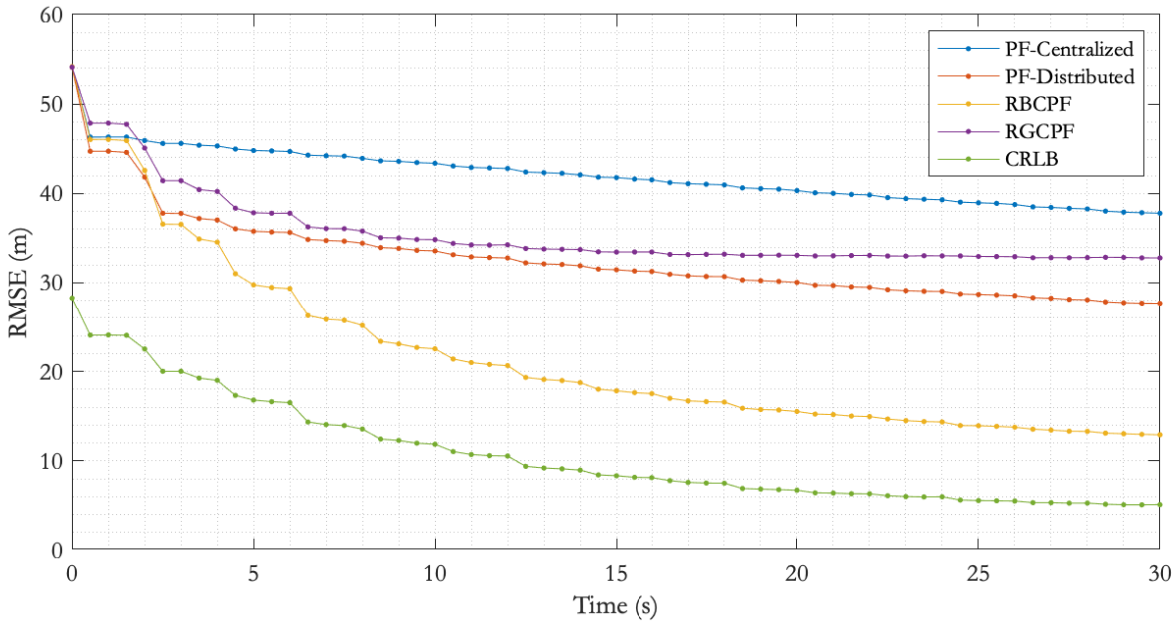


Figure 5–6 RMSE Using CRLB and Particle Filters–150 Particles, 8 Nodes, 500 Runs

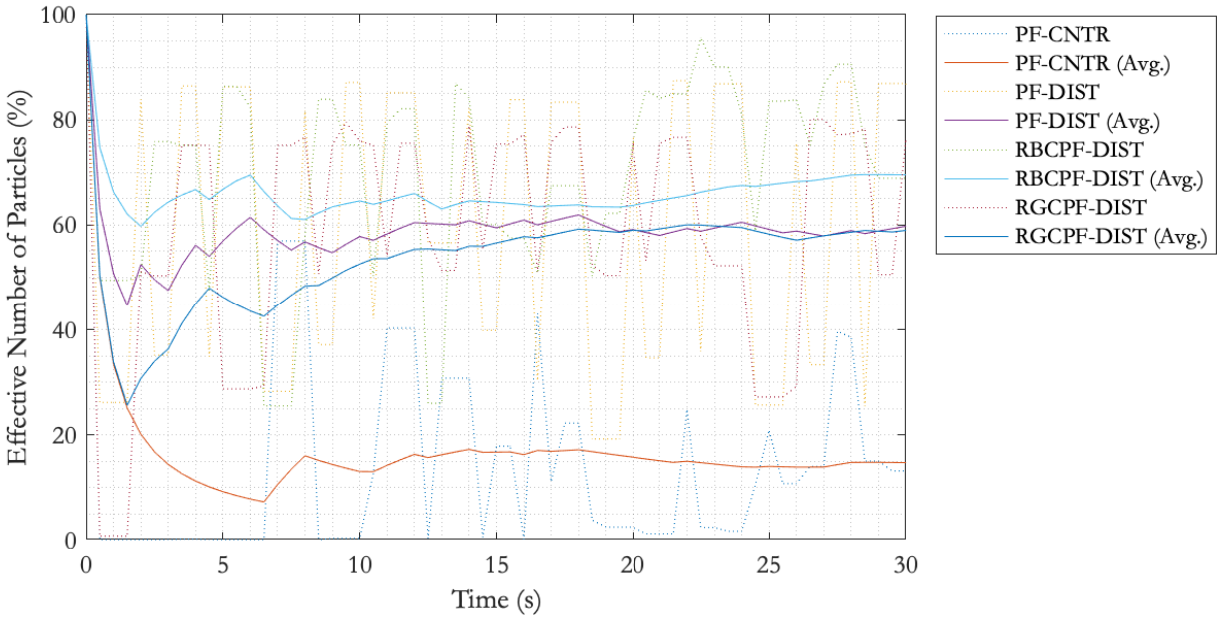


Figure 5–7 Effective Number of Particles, 4 Nodes

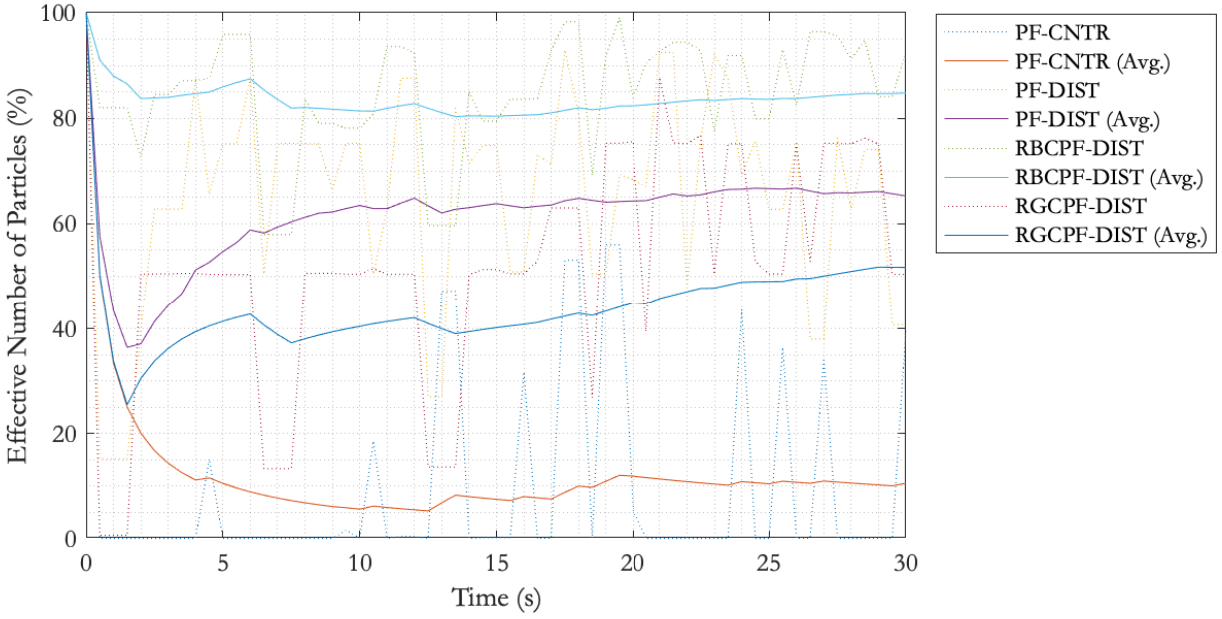


Figure 5–8 Effective Number of Particles, 8 Nodes

Figure 5–9 to Figure 5–12 show the cumulative measurement graph for the four nodes scenario along with the posterior distribution of one of the nodes for the first few measurements updates. The figures show the evolution of the posterior distribution. At time zero, the state is initialized using a uniform a priori distribution, with no active range measurements between nodes. With every new measurement, the posterior is updated. In the cases of the centralized and the distributed particle filter, the particles population collapses rapidly, for the reasons outlined above, slowing the convergence of the filters to the correct state. However, the RBCPF and the RGCPF generally maintain a more diverse particles population, making the convergence faster as more observations become available. The execution time of the different filters in the two scenarios are compared in Table 5–1 and Table 5–2. The RBCPF takes 6-18 times more execution time than the reference centralized filter implementation, while the RGCPF consumes 16-23 times more.

Table 5–1 Relative Run-Time–150 Particles, 4 Nodes

PF Centralized	PF Distributed	RBCPF	RGCPF
1.00x	1.31x	6.24x	16.74x

Table 5–2 Relative Run-Time–150 Particles, 8 Nodes

PF Centralized	PF Distributed	RBCPF	RGCPF
1.00x	1.29x	17.54x	22.50x

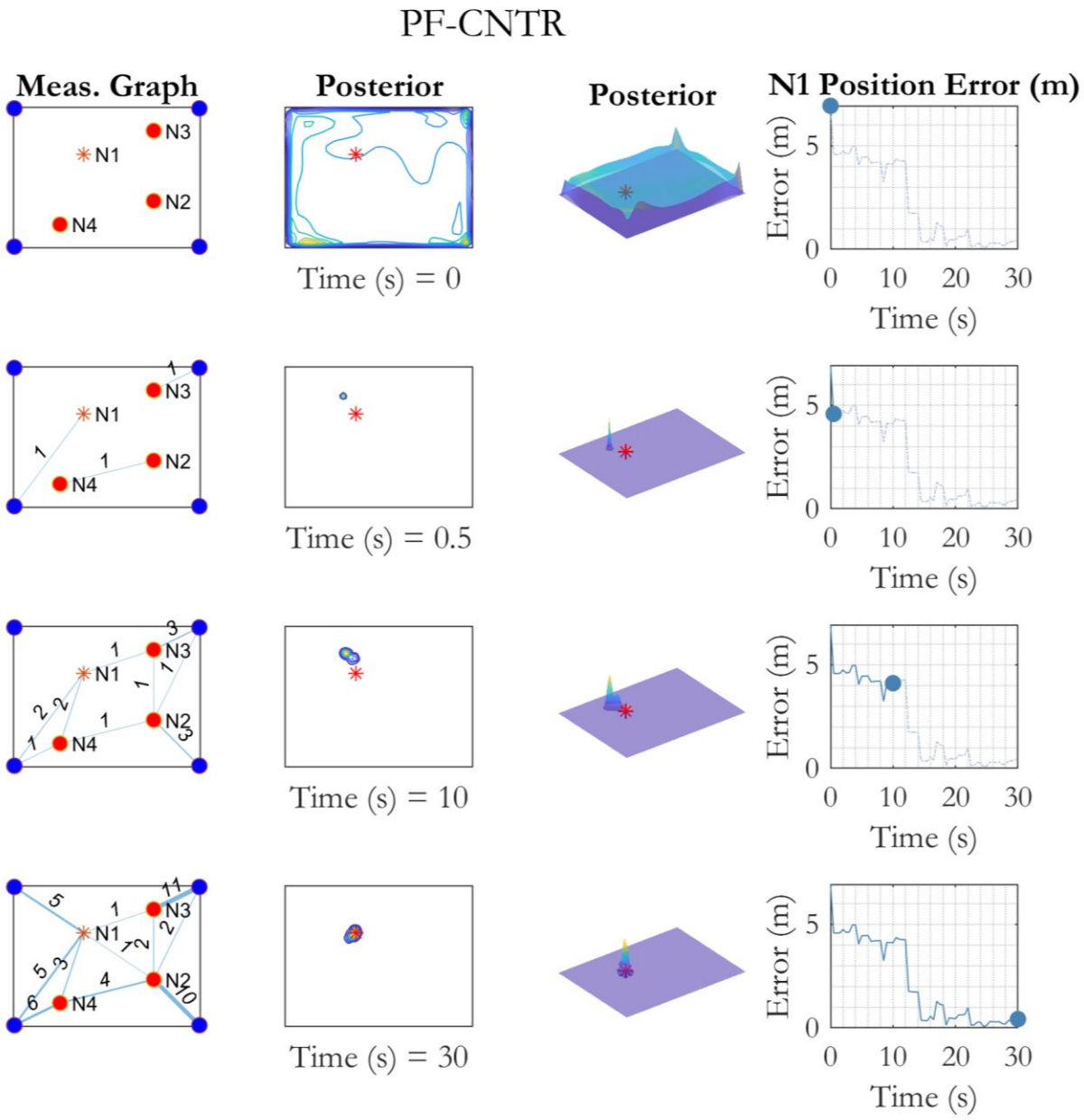


Figure 5–9 Posterior Distribution for Centralized PF, 4 Nodes

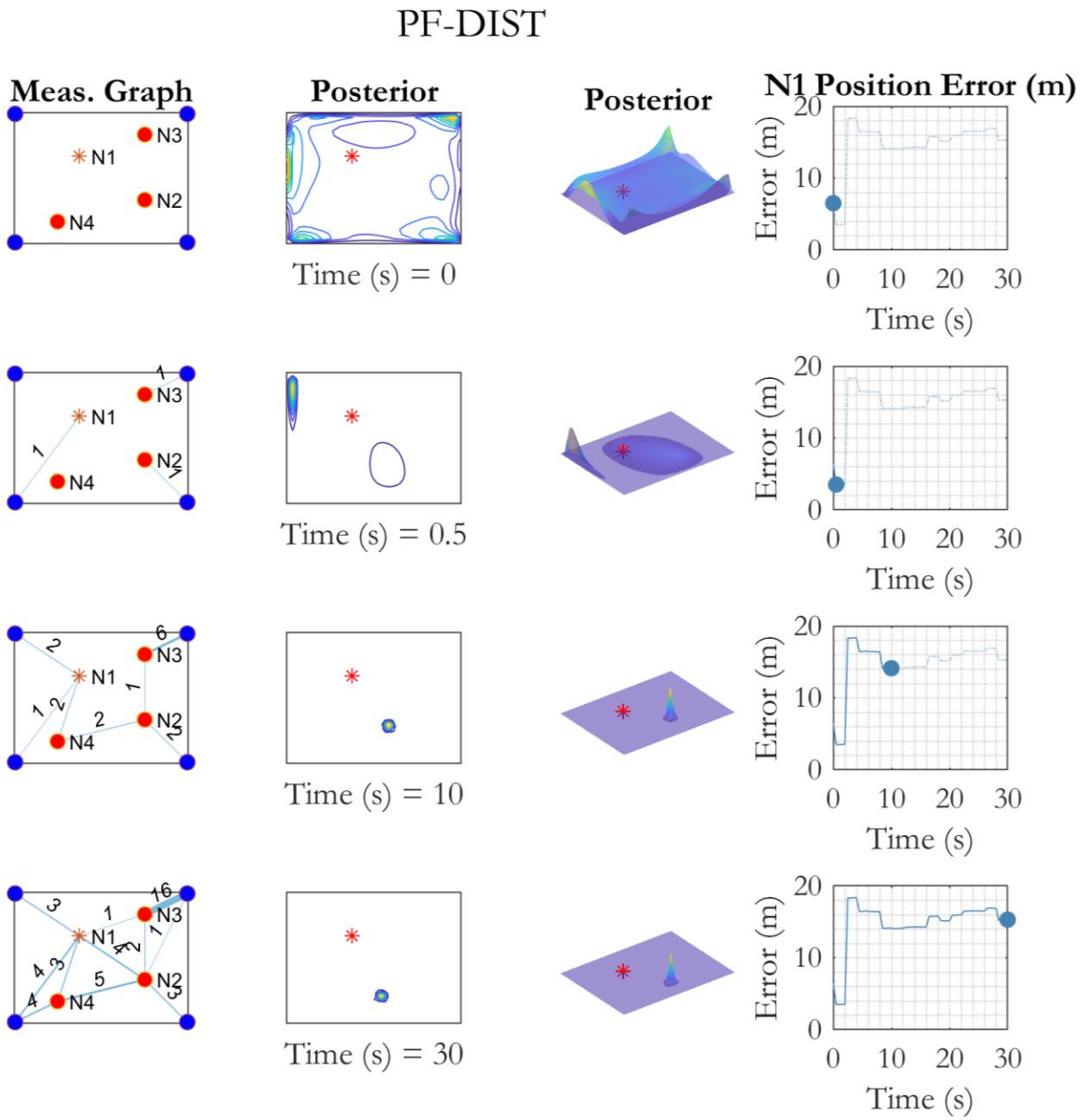


Figure 5–10 Posterior Distribution for Distributed PF, 4 Nodes

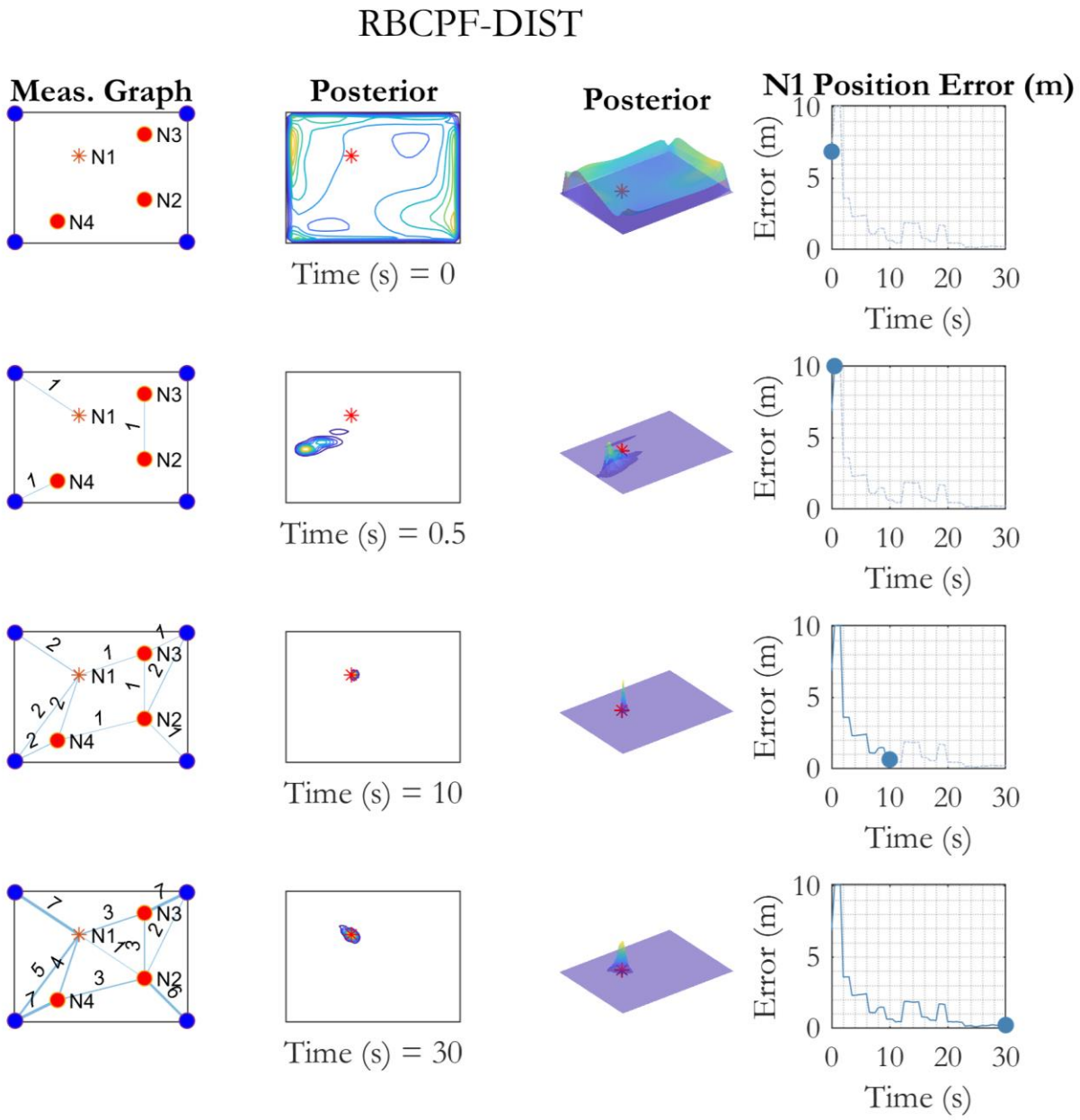


Figure 5–11 Posterior Distribution for RBCPF, 4 Nodes

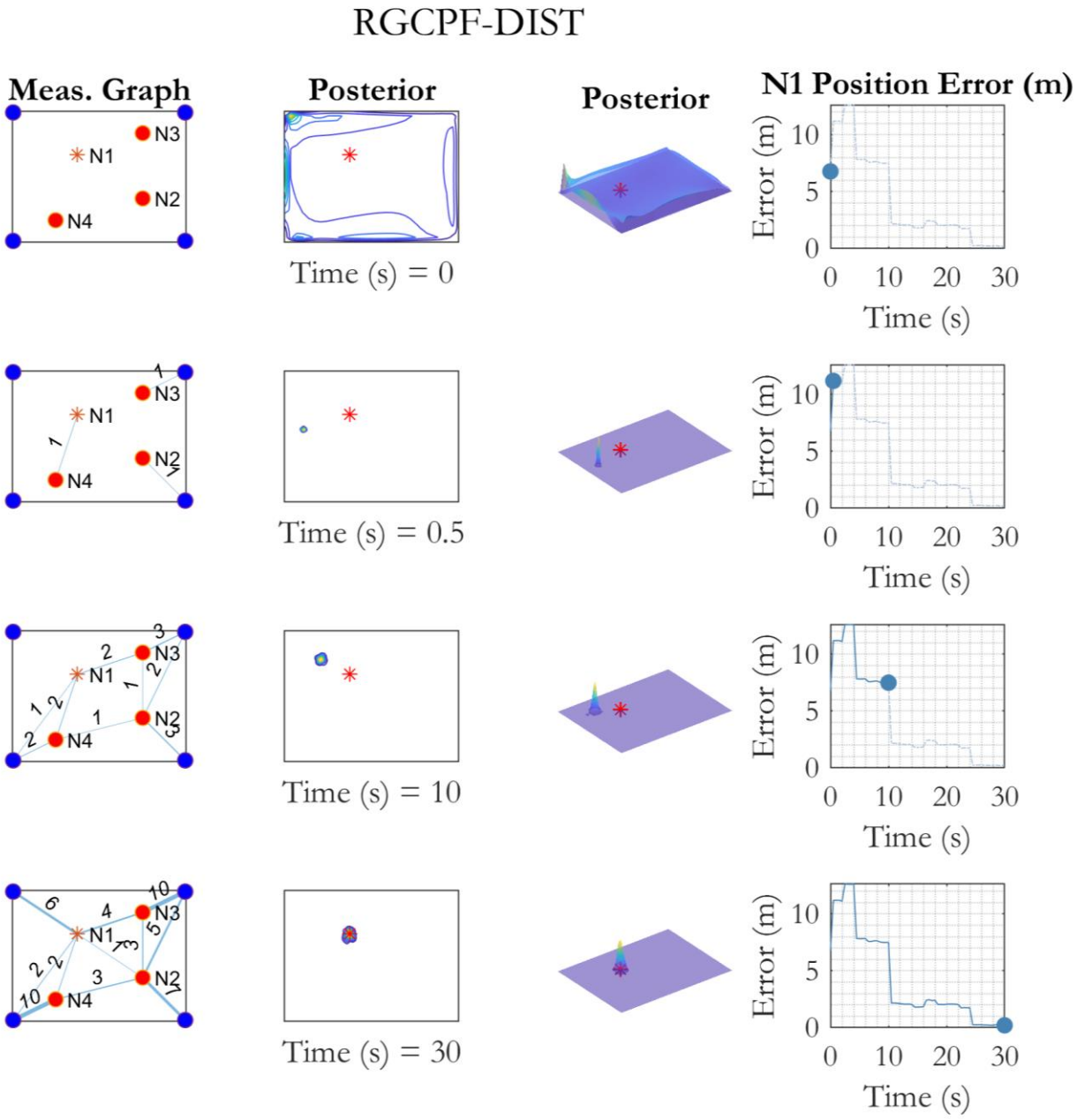


Figure 5–12 Posterior Distribution for RGCPF, 4 Nodes

5.4 Summary

This chapter presented two new collaborative particle filters, using the state marginalization concept to introduce the considered state into the particle filter–similar to the way the original Rao-Blackwellized particle filter handles linear substructures in the state-space model. The Rao-Blackwellized Collaborative Particle Filter (RBCPF) treats the considered state as a conditionally linear random variable, dependant on each particle values. The considered state is updated locally, and the cross-covariance information is calculated and used in later time steps to evaluate the conditional distribution of the considered states. The Rao-Blackwellized Gaussian-importance Collaborative Particle Filter (RGCPF) adds a Gaussian proposal distribution and treats the considered state in a similar way to the Schmidt-Kalman filter.

The performance of the two filters is evaluated against a centralized and distributed particle filters and compared to the RMSE derived from the Cramér-Rao Lower Bound (CRLB). The filters are used to estimate the positions of multiple collaborating nodes in a plan, using intermittent relative and absolute range measurements. The simulations showed that on average, the RBCPF and the RGCPF could achieve better performance compared to the centralized particle filter implementation, with the RBCPF significantly outperforming the centralized and the distributed counterparts. The RBCPF achieve better performance in terms of the achievable RMSE and in terms of achieving a higher percentage of effective particles compared to the centralized and the distributed particle filters.

This chapter defined two vital elements of the proposed framework: the type of data shared between nodes along with an algorithm to use this shared data to update the state of collaborating nodes in a decentralized way.

Chapter Six: **Experiments and Results**

6.1 Experimental Setup and Testing Scenarios

This chapter describes the results of the experimental evaluation of the proposed framework. The experiments took place on the second floor of the engineering block E (ENE) building at the University of Calgary. The surface area of the testing region is 360 m^2, the length of the testing region is 48 m, and the average width is 7.5 m. Figure 6–1 shows the floorplan of the testing area, the locations of the Pozyx fixed anchors, and the locations of the reference points fixed on the floor of the testing area.

The experiments carried out using four dynamic nodes; one of them is the main node, while the other three nodes are the collaborating nodes. This chapter will evaluate the positioning performance of the main node, which was connected to the Pozyx system to collect the ground-truth trajectory. Each node is equipped with a smart device, running the data collection application, and the locSpeck UWB device. Table 6–1 shows the technical details of the smart devices used by the collaborating nodes.

The complete experiment is divided into three separate trajectories. Each trajectory starts with the four nodes at rest. Once the experiment starts, the nodes walk in a random trajectory inside the testing area. The nodes occasionally stop on the reference position during their travel. However, only the reference trajectory of the main node is captured using the Pozyx reference system.

The data logging application runs on each of the smart devices and collects readings from the available sensors, the wi-fi received signal strength indicator along with information about the corresponding access point, and the UWB range measurement along with the address of the collaborating node. This information is processed later, on a desktop computer.

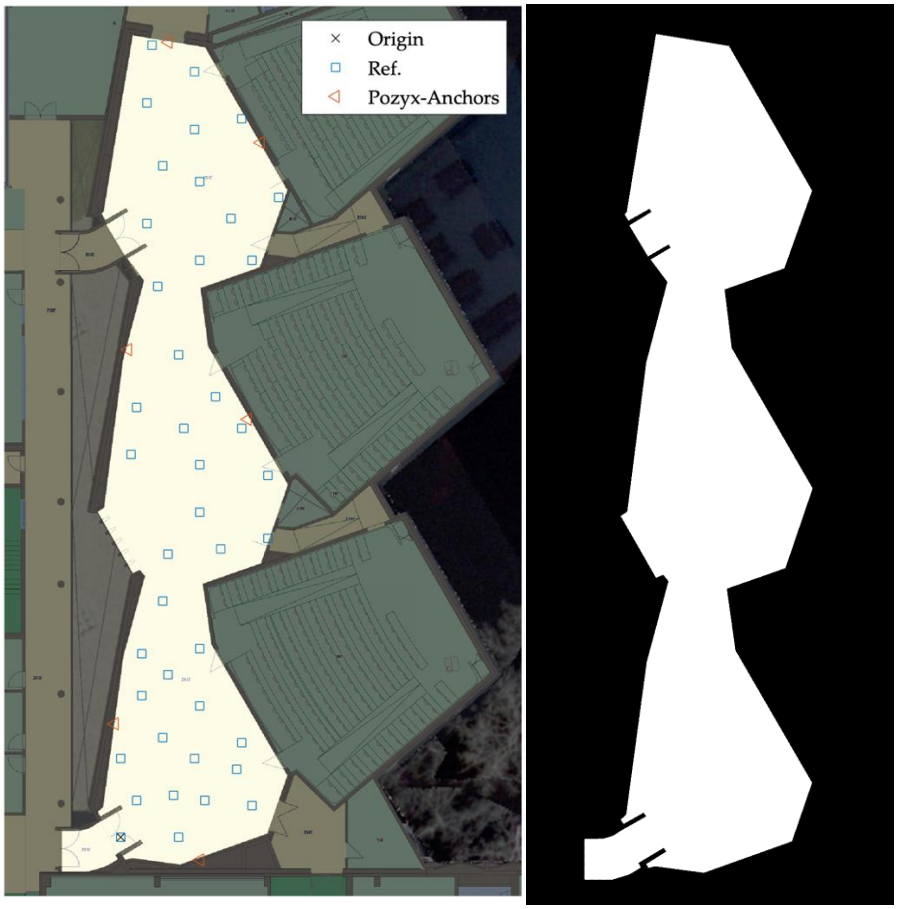


Figure 6–1 Testing environment floorplan

Table 6–1 Technical details of the collaborating nodes

Device	Role	OS	Used Sensors
Nexus 9 (2014)	Main	Android 7.1.1 (3.10.103)	Accelerometer, Gyroscope, Wi-Fi 802.11 a/b/g/n/ac, dual-band
Samsung Galaxy J1 (2015)	Support	Android 6.0.1	Accelerometer, Wi-Fi 802.11 b/g/n
Huawei Nova Plus (2016)	Support	Android 7.0 (3.18.31)	Accelerometer, Wi-Fi 802.11 b/g/n
LG G6 (2017)	Support	Android 7.0 (3.18.31)	Accelerometer, Gyroscope, Wi-Fi 802.11 a/b/g/n/ac, dual-band

The rest of this chapter will discuss the results of the different testing scenarios. The next section evaluates the standalone positioning performance of the main node. In this scenario, all the sensors available to the main node are used by the positioning filter. The Wi-Fi fingerprint map used in this scenario is the reference map, which was created previously using a dedicated trajectory. The objective of this scenario is to establish a performance baseline to which the performance of the collaborative approach is compared.

The following section is dedicated to the evaluation of the collaborative positioning algorithm. In this section, different collaboration scenarios are used. In the first scenario, the main node is not using any of the available sensors, except for the UWB ranging device. While the supporting nodes are estimating their positions using all the sensors available to them, along with the Wi-Fi reference map. The main node uses only the relative range measurements to estimate its position. The objective of this scenario is to evaluate the effect of collaboration in the case of node asymmetry. The main node, in this case, is in a disadvantageous position where it could not estimate its location without external aid from the collaborating nodes.

The second scenario testing for collaborating nodes uses the complete set of sensors on all nodes, main and supporting. The objective of this test is to assess the effect of collaboration when the active node already has a good estimate of its position using only measurements local to the device, without any external sources. A final scenario is tested, where the filter is providing position estimates based on a random-walk model. This final scenario, though seems trivial, is used to establish the performance lower bound.

6.2 Standalone Positioning Scenarios

6.2.1 Positioning Filter Overview

The standalone position algorithm is implemented using a particle filter. The version used for these results uses a pedestrian dead-reckoning (PDR) algorithm for the state update using input from the gyroscope and the accelerometer. The filter uses Wi-Fi fingerprinting to update the weights of the particles, using a Gaussian process model as the reference map. The details of the pedestrian dead-reckoning and the Gaussian process-based fingerprinting were discussed earlier and will not be repeated here. The filter also uses floorplan information to ensure that the effective particles are within the area of interest and eliminate out-of-bound particles. The right panel in Figure 6–1 shows the floor map mask used to discriminate the out-of-bound particles. Equation (6.1) shows the weight update equation using the floorplan information:

$$\widetilde{w}_{\mathrm{i}} = \begin{cases} w_i, & x_i \in FP \\ 0, & x_i \notin FP \end{cases} \tag{6.1}$$

where x_i is the position of the i-th particle, with weight w_i, and FP is the floorplan.

After the weight update step, a weight normalization and resampling steps are implemented to remove the undesired particles. The filter is implemented with relatively low particle count to reduce the processing time required. The positioning filter is the RBCPF with 150 particles.

6.2.2 Reference Trajectories and Fingerprints Maps

The framework performance is evaluated using three different trajectories covering the same test area. Figure 6–2 shows the reference solution for the test trajectories. This reference is created using the Pozyx UWB-based system. The locations of the Pozyx anchors are highlighted in Figure 6–1. The position error is evaluated at any of the pre-surveyed reference points.

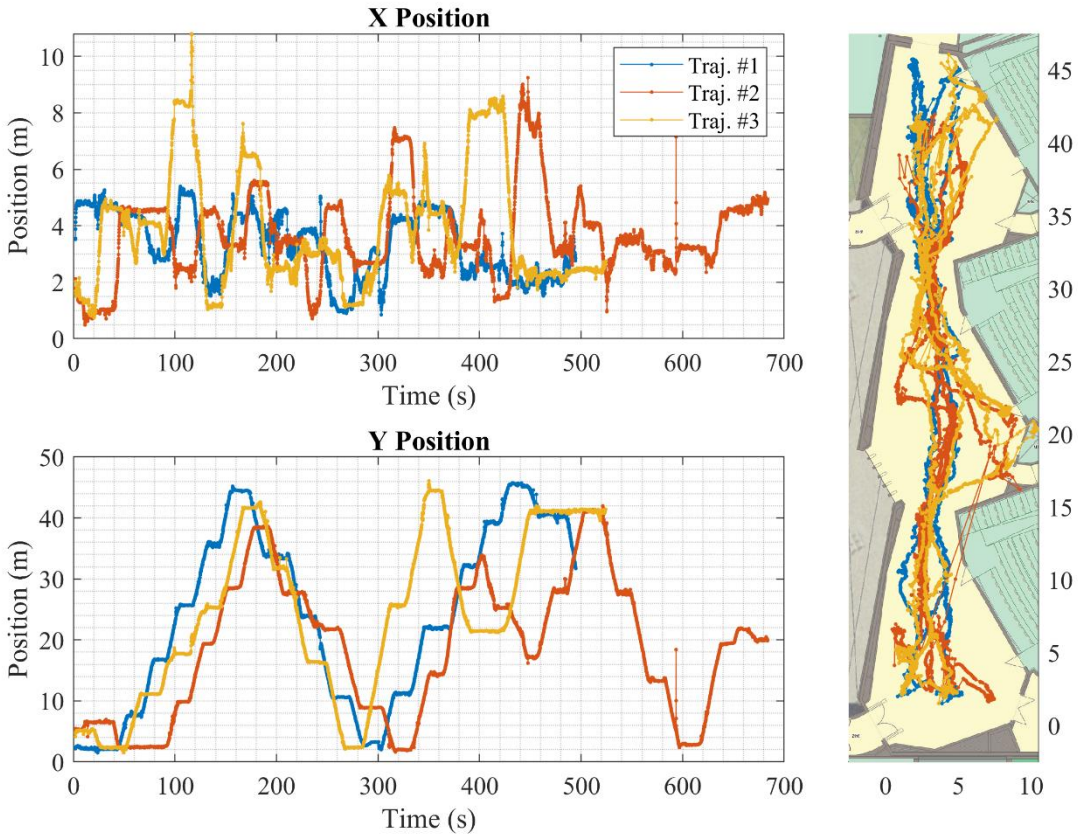


Figure 6–2 Pozyx reference trajectory

Table 6–2 Pozyx positioning error summary

Trajectory #	Mean Absolute Error (m)	RMS Error (m)
1	0.37	0.45
2	0.57	0.65
3	0.60	0.72

The reference solution for each trajectory is compared to the pre-surveyed reference points on the ground. The performance of the Pozyx solution is summarized in Table 6–2. For all the tested scenarios, the position error is evaluated when the node reaches and stops over one of the reference points. This event is captured from the Pozyx reference trajectory in addition to the stop detection algorithm applied to the accelerometer data from the node of interest. The Pozyx

trajectory is not used directly to evaluate the performance. It is used to indicate the location of the nearest reference point on the floor, which location is known precisely, and this reference point is used to evaluate the error in the position estimate. The small positioning error of the Pozyx system is vital to be able to distinguish between the densely-placed reference points.

The reference radio map is created using the Pozyx reference trajectory in a separate run. The fingerprint map is built by observing the signal strength indicator at the reference points, then fit a Gaussian process model, for each visible access point, using the position and signal strength pairs. During the positioning scenarios, the resulting Gaussian process models are used by the different dynamic nodes to aid the positioning filter.

6.2.3 Standalone Positioning Results

Figure 6–3, Figure 6–4, and Figure 6–5 show the positioning results of the main node in the standalone scenario. Each scenario is run through the positioning filter for 20 times, and the results for these runs are overlapping in the figures. The cumulative distribution function of the positioning error is shown in Figure 6–6. The standalone positioning error statistics for the three trajectories are summarized in Table 6–3.

For this scenario, the main node is using all the sensors available onboard the smart device; i.e. gyroscope, accelerometer, and Wi-Fi information. The root-mean-square (RMS) positioning error across the three trajectories ranges from 4.28 m to 6.65 m, while the overall RMS positioning error, in this case, is 5.92 m, as shown in Table 6–7.

Table 6–4 shows the results of the IPIN competition winners from 2015 to 2018 [222], [223]. These results are shown for comparison with the achievable performance of the standalone mode of the framework. The 75% percentile of the position error is not far from the top indoor positioning system available, although the winner of the 2018 off-site track can achieve much

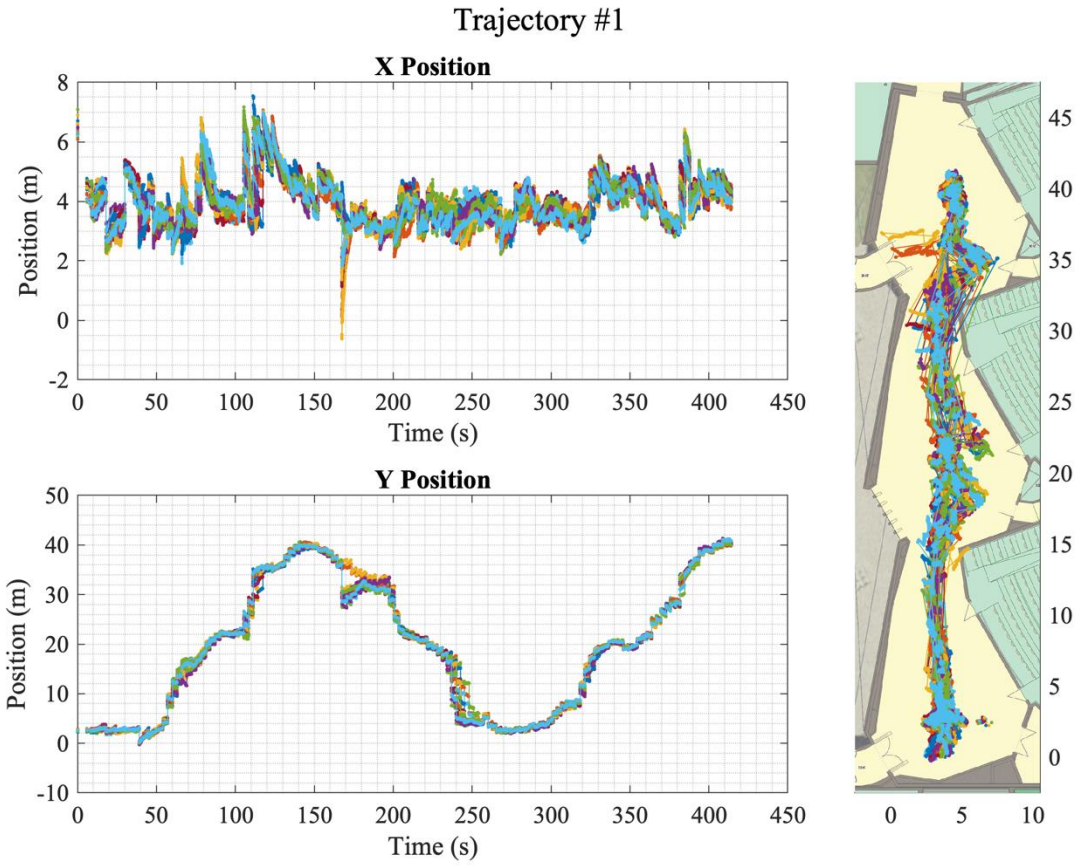


Figure 6–3 Position estimates for trajectory #1

better accuracy. However, the inclusion of the results in Table 6–4 does not imply that the different systems can be compared directly since the performance of any positioning system will vary according to the operating conditions. The sole purpose of showing these results is to give a sense of the performance of current state-of-art systems.

The performance of the standalone solution acts as a baseline to which the collaborative positioning approach is evaluated. The left panel in Figure 6–3, Figure 6–4, and Figure 6–5 show the instantaneous position estimate along the x and the y directions. Generally, the filter in the standalone mode can track the correct trajectory. However, the filter can lose track of the correct

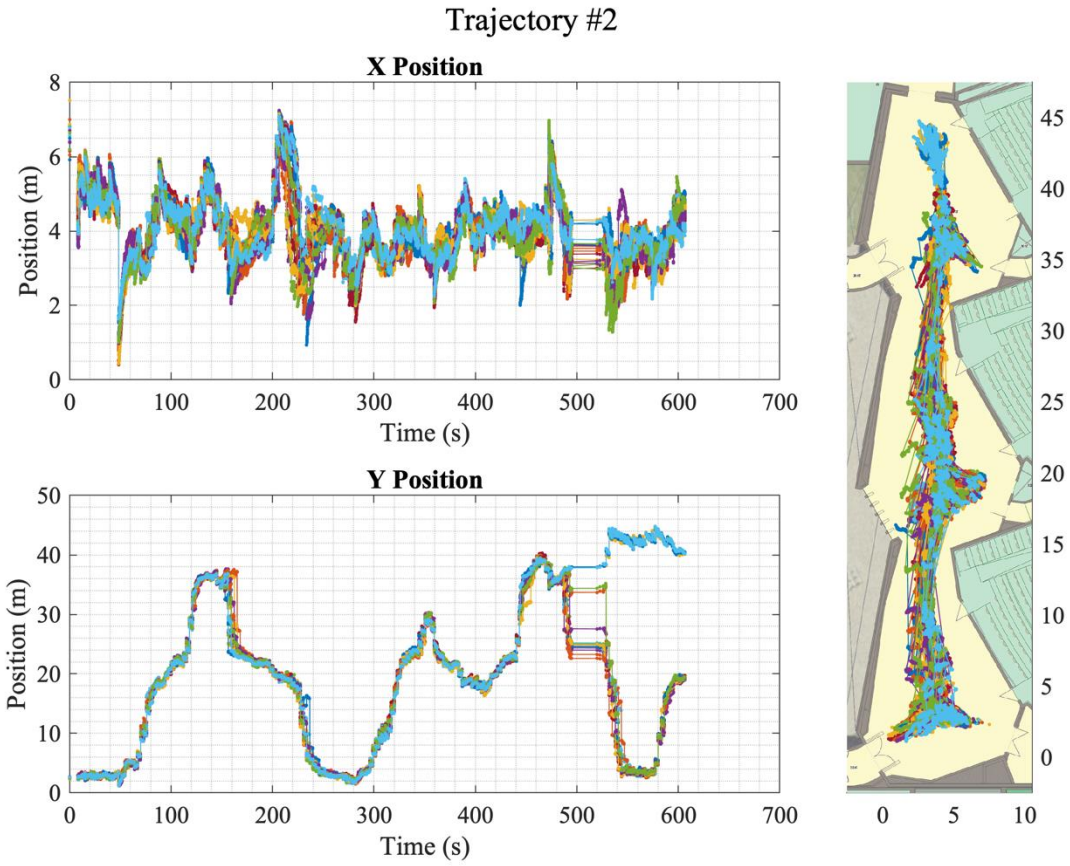


Figure 6–4 Position estimates for trajectory #2

trajectories, as shown in the figures. For example, in Figure 6–4, the filter loses track of the correct trajectory near the end of the trajectory.

Table 6–3 Standalone positioning results summary

Error Stats. (m)	Traj. #1	Traj. #2	Traj. #3	Overall
Mean	3.80	4.33	4.84	4.36
Min	0.27	0.02	0.28	0.02
Max	9.14	40.84	21.85	40.84
50% Percentile	3.34	3.77	4.41	4.09
75% Percentile	5.30	5.62	5.14	5.34
90% Percentile	6.82	7.13	6.57	6.85
RMS	4.28	6.65	5.75	5.92
Std. dev.	1.97	5.04	3.10	4.01

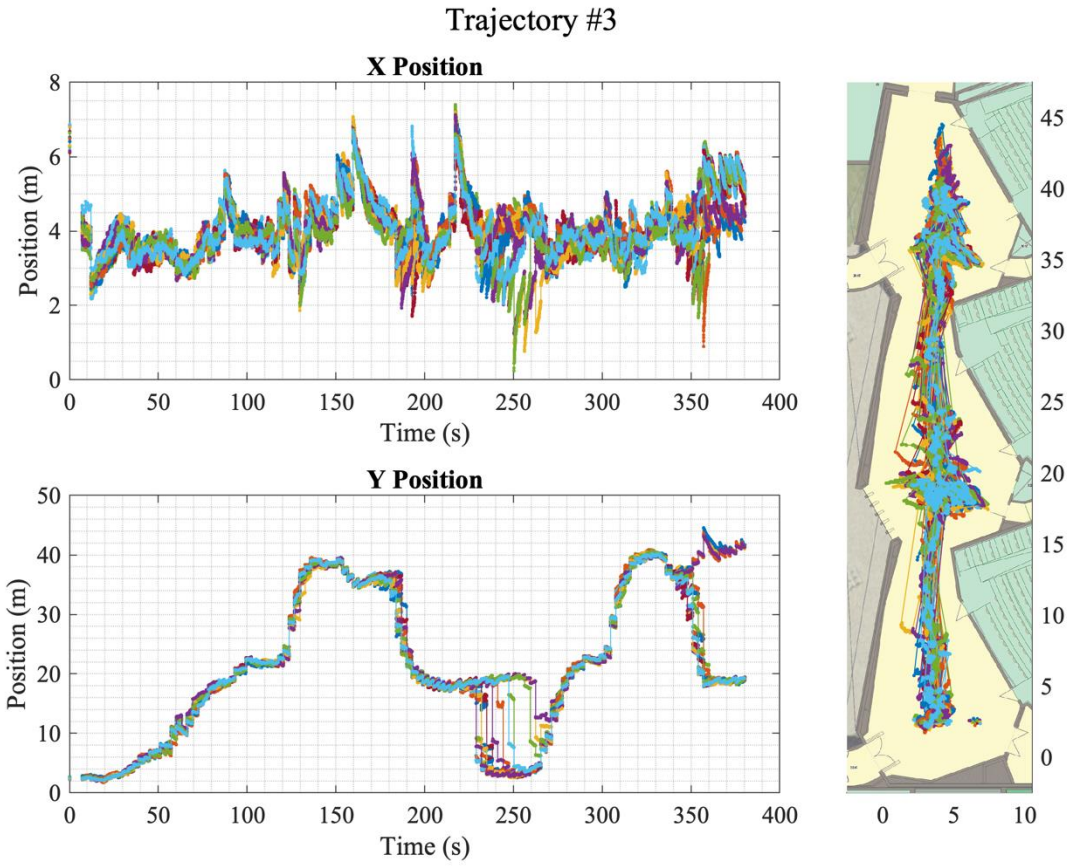


Figure 6–5 Position estimates for trajectory #3

Table 6–4 Performance of indoor positioning competitions (75% percentile)

Competition	Track	Accuracy (m)
IPIN 2015	Smartphone (on-site)	6.6
IPIN 2015	Smartphone (off-site)	8.3
IPIN 2016	Smartphone (on-site)	5.4
IPIN 2016	Smartphone (off-site)	5.8
IPIN 2017	Smartphone (on-site)	8.8
IPIN 2017	Smartphone (off-site)	3.48
IPIN 2018	Non-Camera based Positioning (on-site)	5.5
IPIN 2018	Smartphone (off-site)	1.1

The same behaviour can be noticed in Figure 6–5, where the filter occasionally loses track in the region between 230 s and 280 s. In this case, the filter was able to recover and converge near the

correct trajectory. However, near the end of the trajectory, the filter sometimes diverges and remains at the wrong side of the testing region.

6.3 Collaborative Positioning Scenarios

In this section, two collaboration scenarios are considered. The first scenario consists of four nodes, the main node, and three supporting nodes. The main node will not use any of its onboard sensors. However, the node will use the UWB device to measure the relative ranges between itself and the other collaborating nodes. The other nodes will use all the sensors available to them, along with the range measurement device. The objective of this scenario is to evaluate the achievable performance using relative range measurements to dynamic nodes. The second scenario is similar to the first one with one change; the main node will be using all the sensors available onboard, in addition to the range measurement device. The objective of this scenario is to assess the effect of the collaboration on the participating nodes.

6.3.1 Performance using Random-Walk Model

Before proceeding to evaluate the performance of the collaborative approach using relative range measurements, it would be useful to consider the error in the absence of the collaboration between the main node and the other nodes.

Figure 6–7 shows the CDF for this case, while Table 6–7 shows the positioning error statistics in this case. Without collaboration, the mean of the position error is 18.46 m, while the RMS of the position error is 21.60 m. The 75% percentile of the error is 26.57 m.

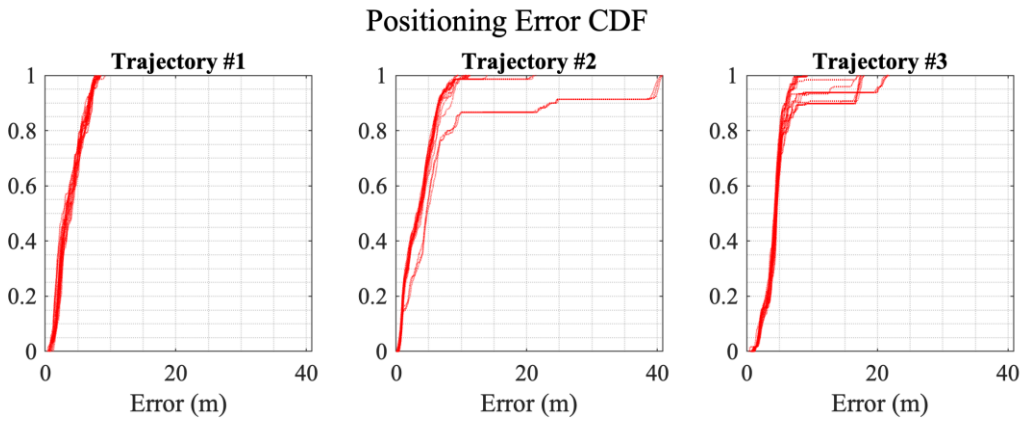


Figure 6–6 Positioning error CDF for standalone scenario

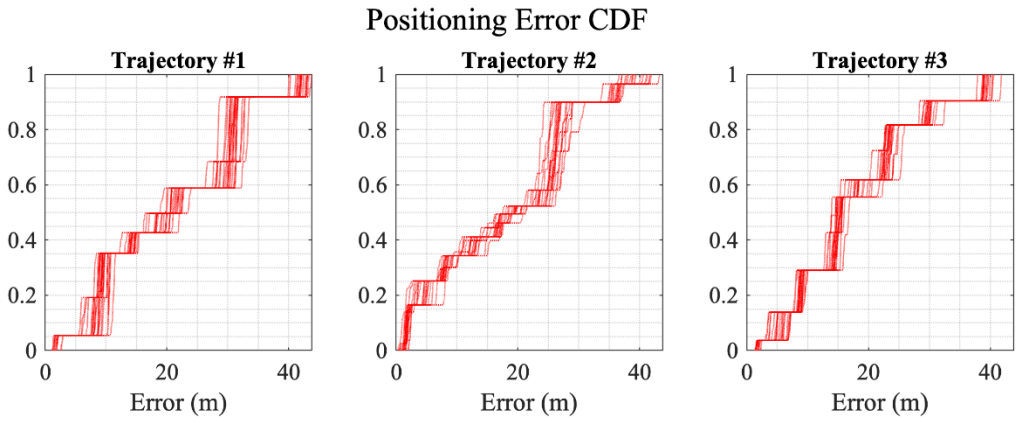


Figure 6–7 Positioning error CDF for the random-walk model

6.3.2 Collaborative Positioning using Relative Range Measurements

The estimated position for the three test trajectories are shown in Figure 6–8, Figure 6–9, and Figure 6–10. These figures show the two-dimensional position estimate along with the instantaneous position in the x and the y directions. Figure 6–11 shows the CDF of the positioning error for each trajectory.

Table 6–5 shows the performance summary for the collaborative positioning approach, using the relative range measurement only. As expected, the performance, in this case, is worse than the performance of the standalone case. However, in this scenario, the mobile node is using only the range measurements, without any of the onboard sensors. In his case, the use of the collaborative positioning framework improves the positioning error for the main node by 50%.

The run-time behaviour of the filter is shown in Figure 6–8, Figure 6–9, and Figure 6–10. In this case, the filter diverges more frequently than the standalone case. Additionally, the estimated trajectory is more prone to sudden jumps. This behaviour is expected while positioning using relative range measurements. Especially that the collaborating nodes are not fixed, with relatively low ranging rate, and there is no guarantee that the geometry between the different nodes will result in a reasonable position estimate.

Another factor that can affect the performance of the main node in the collaborative setting is the availability of the supporting nodes. The nodes availability is illustrated in Figure 6–12, where each horizontal line represents the activity of the corresponding node. The gaps in the lines indicate that the node is not active. Although there are three supporting nodes, only two of them are active most of the time, and the third is fluctuating between the active and inactive state. The effect of the node availability is evident in the second trajectory, which has the most significant errors among the three trajectories.

6.3.3 Collaborative Positioning using All Sensors

This scenario evaluates the effect of the collaboration on the main node while using the full set of sensors available on board. Using the full sensors, the main node should achieve a performance level similar to the performance of the standalone solution. Figure 6–13 shows the CDF of the positioning error for this case, for the three trajectories.

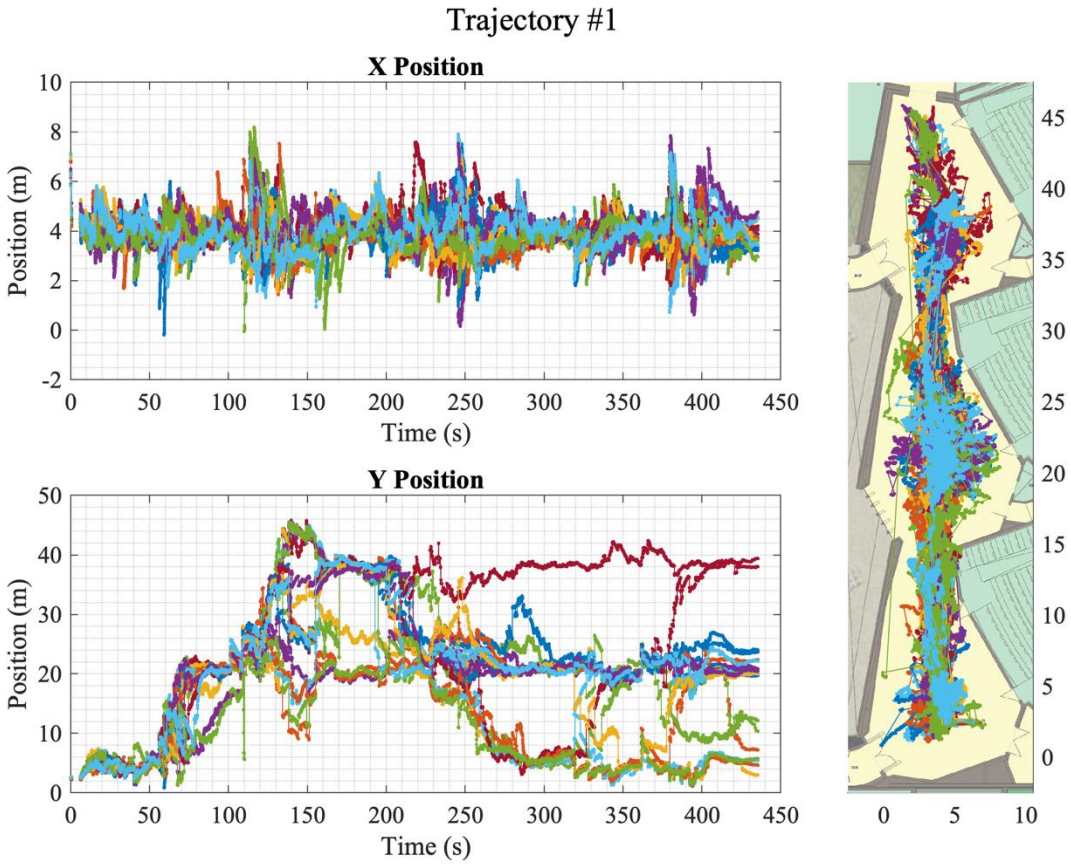


Figure 6–8 Position estimates for trajectory #1

Table 6–5 Collaborative positioning results summary (no sensors)

Error Stats. (m)	Traj. #1	Traj. #2	Traj. #3	Overall
Mean	8.58	10.65	8.36	9.51
Min	0.02	0.00	0.06	0.00
Max	34.61	43.21	27.44	43.21
50% Percentile	6.49	6.99	6.70	6.79
75% Percentile	10.91	16.48	12.89	13.38
90% Percentile	15.90	23.80	16.77	22.42
RMS	10.57	14.32	10.15	12.43
Std. dev.	6.18	9.58	5.77	7.99

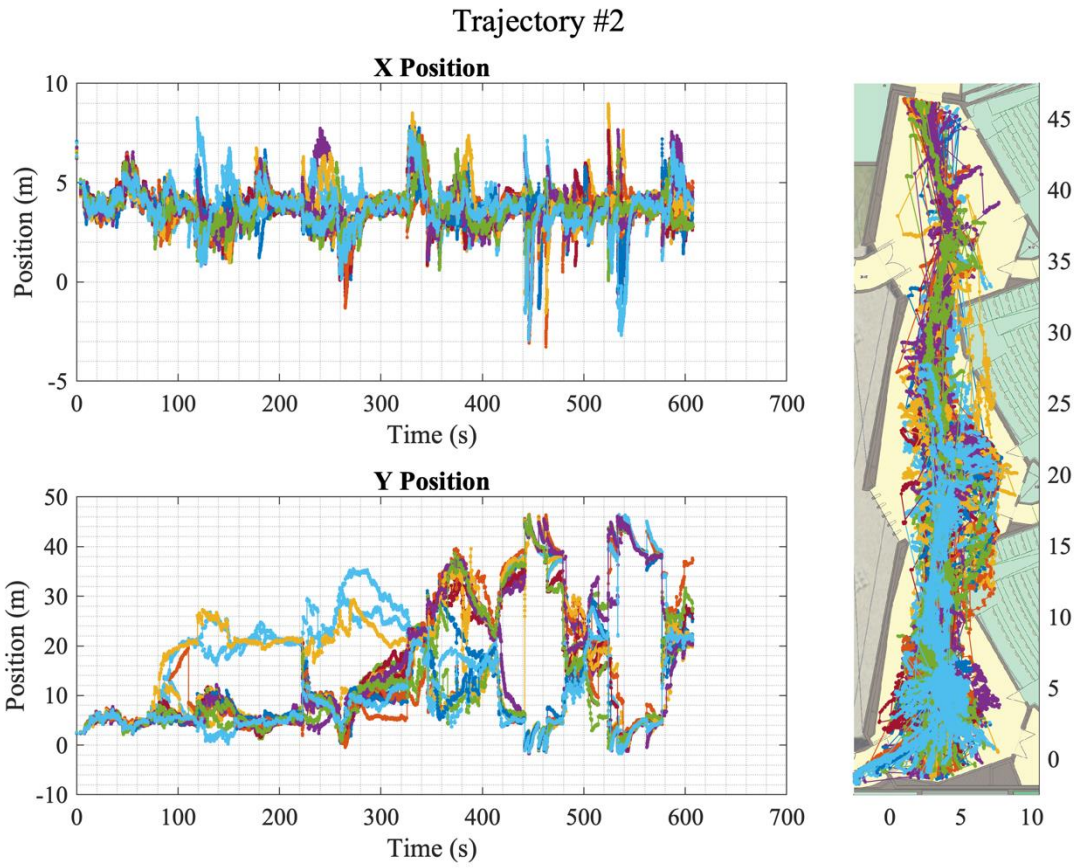


Figure 6–9 Position estimates for trajectory #2

Table 6–6 Collaborative positioning results summary (all sensors)

Error Stats. (m)	Traj. #1	Traj. #2	Traj. #3	Overall
Mean	5.98	5.36	5.49	5.54
Min	0.03	0.01	0.01	0.01
Max	27.26	42.35	23.08	42.35
50% Percentile	4.44	4.96	4.84	4.81
75% Percentile	8.97	6.94	6.35	7.06
90% Percentile	11.16	10.00	9.02	10.36
RMS	7.30	6.92	6.54	6.90
Std. dev.	4.18	4.37	3.55	4.11

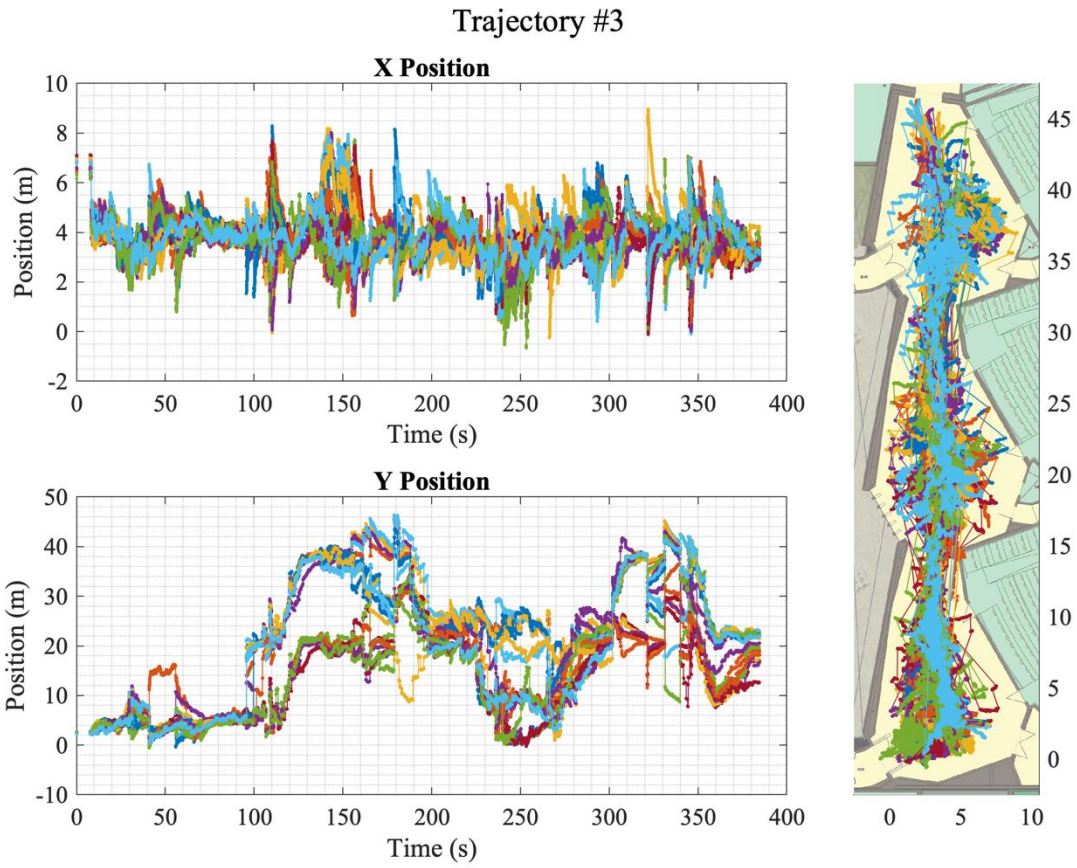


Figure 6–10 Position estimates for trajectory #3

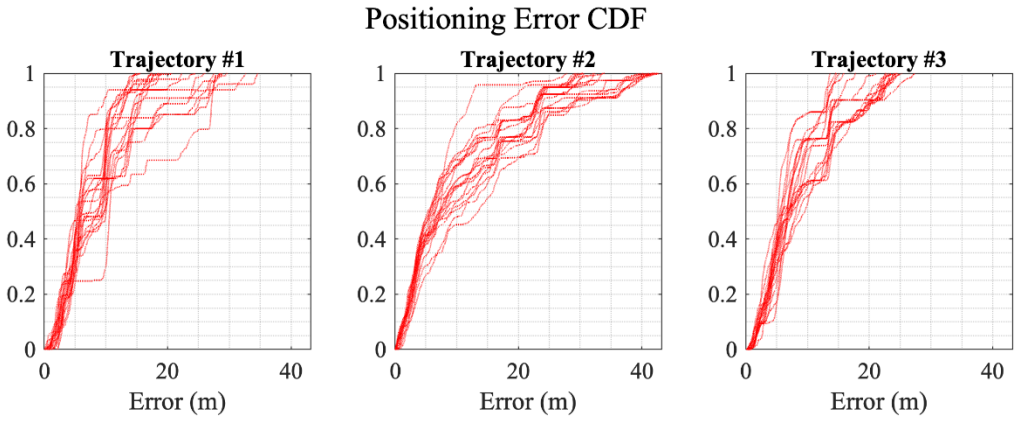


Figure 6–11 Positioning error CDF for collaborative positioning without using sensors on the main node

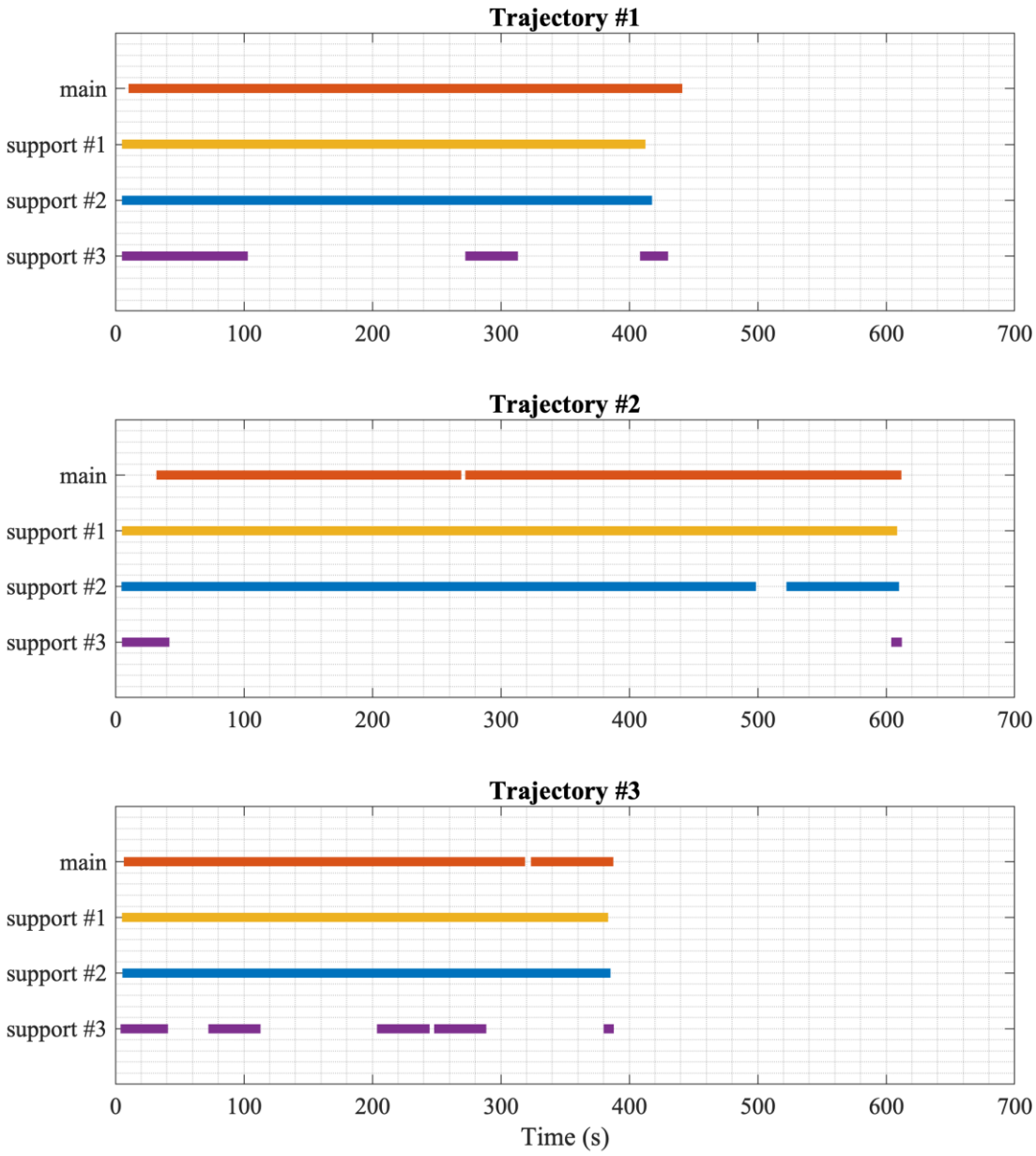


Figure 6–12 Node activity graph

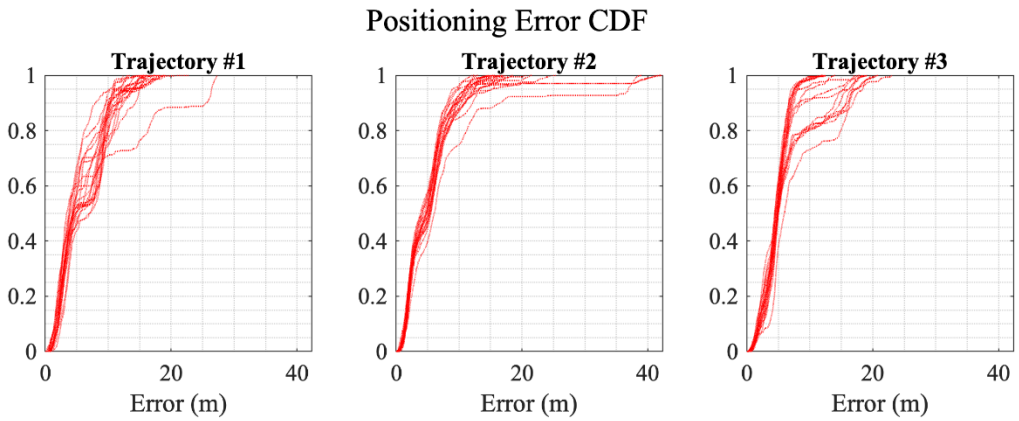


Figure 6–13 Positioning error CDF for collaborative positioning while using the full sensors set on the main node

Table 6–6 shows a summary of the positioning performance of the collaborative positioning, while the main node is using all its sensors. The collaboration negatively affected the performance of the main node, when it uses all the sensors. The mean error has increased by 27%, the RMS error by 16.6%, and the 75% percentile error by 32.2%.

One possible explanation of the performance degradation could be related to the fusion of an erroneous estimate from one of the collaborating nodes. This case is exacerbated when the filter of a collaborating node suffers from particle depletion. As a result, the filter will generate new particles through resampling. These particles will have less diversity, and the corresponding particles covariance will be small. When a filter with these characteristics collaborates with another, it will provide false confidence in its position estimate. Consequently, it will drive the estimates of the collaborating nodes in the wrong direction. The performance degradation due to an erroneous state estimate or an overconfident remote note could be mitigated by implementing filter integrity measures to ensure that each filter has a realistic covariance estimate, possibly by utilizing the actual values of the measurement likelihood function.

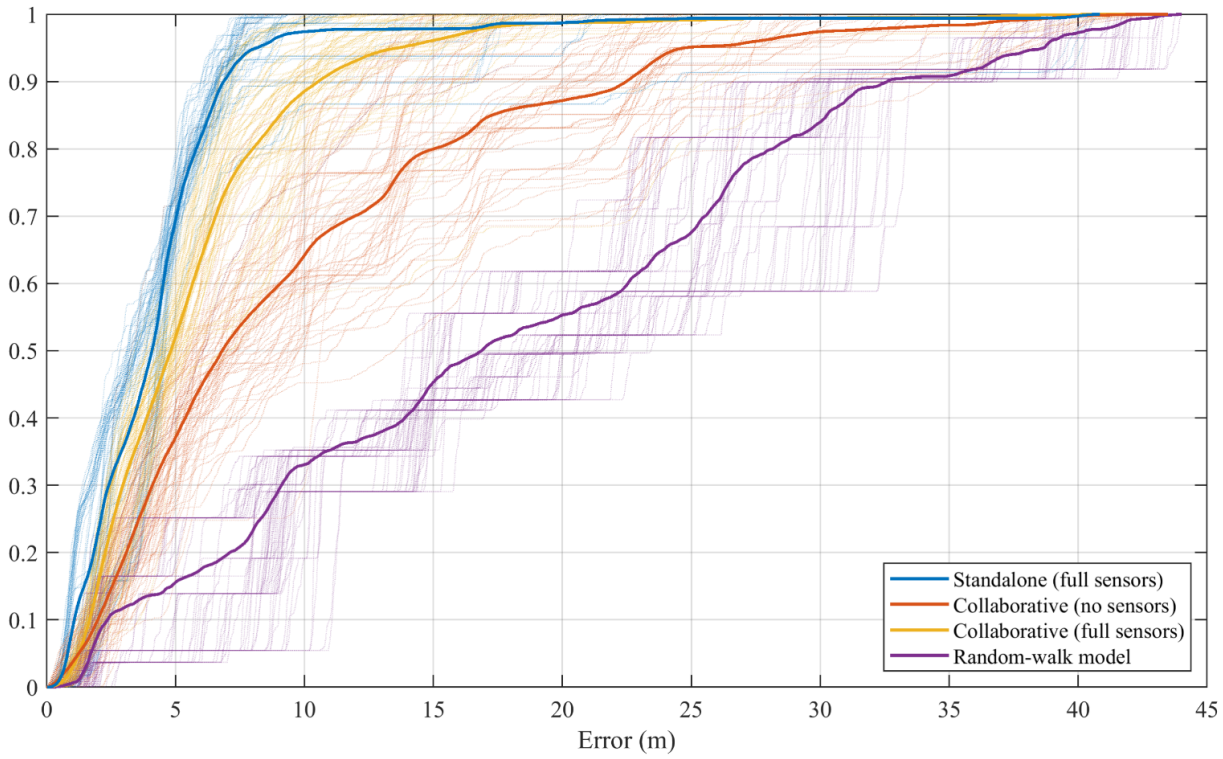


Figure 6–14 The combined positioning error CDF for the four scenarios

Table 6–7 Positioning Results Summary

Error Stats. (m)	Standalone (full sensors)	Collaborative (full sensors)	Collaborative (no sensors)	Random-walk
Mean	4.36	5.54	9.51	18.46
Min	0.02	0.01	0.00	0.25
Max	40.84	42.35	43.21	43.79
50% Percentile	4.09	4.81	6.79	17.89
75% Percentile	5.34	7.06	13.38	26.57
90% Percentile	6.85	10.36	22.42	32.63
RMS	5.92	6.90	12.43	21.60
Std. dev.	4.01	4.11	7.99	11.2

6.4 Summary

This chapter provided an overview of the positioning performance using the proposed collaborative framework. Figure 6–14 shows the cumulative distribution function of the position error of the main node in the four scenarios described earlier in this chapter. Table 6–7 list the overall performance for each of the tested scenario.

It is evident that using the collaborating framework provides a significant advantage to nodes with little or no sensors. Without fusing information from the collaborating nodes, these weak nodes will not be able to estimate its position.

However, for the strong nodes in the process, the performance can suffer a hit. This performance loss can be alleviated by implementing measures to improve the integrity of the filter, such as divergence monitoring [224], or by increasing the number of the particles to better resembles the posterior of the filter.

Chapter Seven: **Conclusions and Future Work**

7.1 Conclusions

This thesis proposed and evaluated a unified collaborative and multimodal framework for indoor positioning and mapping using smartphones. This objective is achieved by evaluating the different building blocks of the framework and improving each of these building blocks to enhance the overall performance and efficiency of the system. The thesis investigated different standalone filtering approaches for the use with environment maps modelled as Gaussian processes. Then, the thesis moved to the mapping task, focusing on developing an efficient representation of the environment maps. The collaboration between the different nodes is achieved by an ad-hoc and dynamic ranging network. To this end, a ranging device is developed and evaluated.

Different simulations and experiments were performed to evaluate the individual components of the framework, along with the whole framework. These experiments confirm the advantage of the proposed framework as discussed earlier in the corresponding chapters, and are outlined in the following section. The outcomes of this thesis prove that using collaborative positioning approaches can provide pronounceable improvement in the performance, especially for situations involving asymmetric nodes, where the weak nodes can benefit from the superior sensing or computational power available in nearby nodes.

7.2 Contributions Summary

The objective of this thesis was to propose and evaluate a new collaborative and multimodal framework for indoor positioning using smartphones. This objective is achieved by proposing and evaluating an indoor collaborative positioning framework that uses relative range measurements and environmental features modelled as Gaussian process models. The relative

range measurement is based on UWB ranging, and the environmental feature is the Wi-Fi signal strength fingerprint.

The thesis defined and evaluated the different building blocks of the proposed framework, while the effectiveness of the entire framework is evaluated through a combination of simulations and real-world experiments.

The major contributions of this thesis can be summarized as follows:

1. *Evaluate different standalone filter architecture for nodes localization*

   The abundance of Wi-Fi access point in indoor environments makes it a practical candidate for aiding localization algorithms. One technique for localization using Wi-Fi signals is fingerprinting of the Wi-Fi received signal strength indicator (RSSI), modelled by Gaussian process (GP) models. The thesis proposes and evaluates different filter architectures to utilize the Wi-Fi RSSI information, along with inertial sensors available on the smartphones. Different kinematic models and update architectures were evaluated in terms of the positioning accuracy, the run-time, and the convergence characteristics of each filter.

2. *Design and evaluate a light-weight Wi-Fi fingerprint map representation using sparse Gaussian process models*

   The environmental feature maps modelled using Gaussian processes are the core of the proposed framework, e.g. the Wi-Fi fingerprint maps. Gaussian process (GP) models are non-parametric models used to evaluate the likelihood function of the Wi-Fi RSSI values at every point on the floorplan, conditioned on a set of training points. A new sparse GP representation is presented to increase the computation efficiency and reduce the storage requirements of the Wi-Fi RSSI maps. The proposed representation is called the Parametric Grid Sparse GP (PGSGP) model. PGSGP reduces the complexity of evaluating the likelihood

function, by reducing the effective number of points in the training dataset, without significant loss of the mapping or positioning accuracy. The PGSGP model is constructed by, first, finding a set of pseudo-inputs arranged over a parametric grid, then optimizing the corresponding target values and the GP model hyperparameters.

3. *Propose a family of distributed particle filters for collaborative positioning*

   The collaborative particle filters are designed to enable using relative measurements between different nodes using a distributed estimation approach, alleviating the need for a centralized fusion computer. Positioning with range measurements in indoor environments is affected by anchor occlusion, weak or unreliable signals caused by multipath effects, or poor geometry, rendering positioning using range measurements to anchors unachievable. This thesis provides a detailed derivation of the filtering equation for the new collaborative particle filter family along with its performance evaluation. The proposed filter family poses no restrictions on the connectivity of the nodes nor assumes the immutability of the connectivity structure. The nodes share only an approximation of the locally calculated posterior distribution and the relative range measurements. The algorithm is tested in a multi-node positioning environment using relative and absolute range measurements.

4. *Implement and evaluate UWB ranging device and the associated logging software*

   The collaboration between nodes in the proposed framework is contingent on the availability of relative range measurements between different nodes. This thesis describes the software and the hardware components of a collaborative positioning system based on UWB transceiver working in an ad-hoc and dynamic network setting. The ranging measurement component of the system is based on the DW1000 UWB transceiver chip from Decawave. The hardware component of the system is a standalone ranging device which is composed of

a UWB transceiver and a Bluetooth Low Energy (BLE) interface to pair the ranging device to a smartphone. The software component of the system comprises the firmware of the ranging device and an Android logging application running on the smartphone. The described system supports real-time addition and removal of nodes, and dynamic node role assignment, either as an anchor or as a rover. The performance of the system is evaluated by a real-world test using a group of four devices navigating an indoor environment.

7.3 Future Work

The proposed framework is designed to be as flexible and expandable as possible. The following lines suggest some directions for improving and extending the functionality of the proposed framework:

- Improve the performance of the standalone positioning algorithm by incorporating more observations and estimating the biases and parameters of the model in real-time.
- Include more environmental features maps; such features include the ambient magnetic field intensity, the ambient light intensity, or visual landmarks.
- Investigate methods to improve the efficiency of the proposed ad-hoc peer-to-peer network by utilizing multiple radio channel simultaneously or by using different preamble codes.
- Build a custom board for the locSpeck ultra-wideband ranging device to reduce the overall size of the device and increase its applicability.
- Evaluate the framework using 5G hardware for indoor and outdoor applications.
- Extend the framework for outdoor navigation and positioning by including GNSS observations, cell towers information, or the geomagnetic field information.
- Enhance the integrity of the distributed positioning algorithm by, for example, implementing methods to monitor the particle divergence in the particle filter.

- Migrate the framework codebase to a light-weight and efficient language, and optimize its performance for real-time applications, whether targeting smart device or embedded processors.
- Investigate the use of simultaneous localization and mapping algorithms within the framework, using the proposed efficient map representation. The ability to collaboratively build a map of the different environmental feature is a key extension of the current framework.

Chapter Eight: **References**

[1] Z. Chaloupka, "Technology and Standardization Gaps for High Accuracy Positioning in 5g," *IEEE Communications Standards Magazine*, vol. 1, no. 1, pp. 59–65, Mar. 2017.

[2] Markets and Markets, "Location Based Services Market by Type & Software - 2023," Jul-2018. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/location-based-service-market-96994431.html. [Accessed: 30-Mar-2019].

[3] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[4] C. Fischer and H. Gellersen, "Location and navigation support for emergency responders: A survey," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 38–47, 2010.

[5] R. Harle, "A Survey of Indoor Inertial Positioning Systems for Pedestrians," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.

[6] A. Correa, M. Barcelo, A. Morell, and J. L. Vicario, "Enhanced Inertial-Aided Indoor Tracking System for Wireless Sensor Networks: A Review," *IEEE Sensors Journal*, vol. 14, no. 9, pp. 2921–2929, Sep. 2014.

[7] A. K. M. M. Hossain and W.-S. Soh, "A Survey of Calibration-Free Indoor Positioning Systems," *Computer Communications*, vol. 66, pp. 1–13, Jul. 2015.

[8] J. Shang, X. Hu, F. Gu, D. Wang, and S. Yu, "Improvement Schemes for Indoor Mobile Location Estimation: A Survey," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–32, 2015.

[9] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu, “Mobility Increases Localizability: A Survey on Wireless Indoor Localization Using Inertial Sensors,” *ACM Comput. Surv.*, vol. 47, no. 3, pp. 54:1–54:34, Apr. 2015.

[10] S. He and S.-H. G. Chan, “Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[11] A. Yassin *et al.*, “Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1327–1346, Second-quarter 2017.

[12] P. Davidson and R. Piché, “A Survey of Selected Indoor Positioning Methods for Smartphones,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1347–1370, Second-quarter 2017.

[13] C. Wu, Z. Yang, and Y. Liu, *Wireless Indoor Localization: A Crowdsourcing Approach*, 1st ed. 2018 edition. New York, NY: Springer, 2018.

[14] B. Jang and H. Kim, “Indoor Positioning Technologies Without Offline Fingerprinting Map: A Survey,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 508–525, First-quarter 2019.

[15] Y. Liu and Z. Yang, *Location, localization, and localizability: location-awareness technology for wireless networks*. New York: Springer, 2011.

[16] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Intelligent Robotics and Autonomous Agents : Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, US: The MIT Press, 2011.

[17] Y. Li, Y. Zhuang, H. Lan, Q. Zhou, X. Niu, and N. El-Sheimy, "A Hybrid WiFi/Magnetic Matching/PDR Approach for Indoor Navigation with Smartphone Sensors," *IEEE Communications Letters*, vol. 20, no. 1, pp. 169–172, Jan. 2016.

[18] Y. Li, "Integration of MEMS Sensors, WiFi, and Magnetic Features for Indoor Pedestrian Navigation with Consumer Portable Devices," University of Calgary, 2016.

[19] P. Robertson *et al.*, "Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, 2013, pp. 1–10.

[20] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1028–1035, Oct. 2009.

[21] A. Bensky, *Wireless positioning technologies and applications*. Boston, Mass: Artech House, 2008.

[22] P. Misra and P. Enge, *Global Positioning System, Signals, Measurements, and Performance*, Rev. 2nd ed. Ganga-Jamuna Press, 2011.

[23] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, *WLAN Positioning Systems*. Cambridge University Press, 2012.

[24] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, no. 8, pp. 57–66, 2001.

[25] C. Yang and H.-R. Shao, "WiFi-Based Indoor Positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, Mar. 2015.

[26] J. B. Andersen, T. S. Rappaport, and S. Yoshida, “Propagation measurements and models for wireless communications channels,” *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, Jan. 1995.

[27] Z. Yang, Z. Zhou, and Y. Liu, “From RSSI to CSI: Indoor Localization via Channel Response,” *ACM Comput. Surv.*, vol. 46, no. 2, pp. 25:1–25:32, Dec. 2013.

[28] Xuyu Wang, Lingjun Gao, Shiwen Mao, and S. Pandey, “DeepFi: Deep learning for indoor fingerprinting using channel state information,” in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, 2015, pp. 1666–1671.

[29] Y. Zhuang, “Integration of WiFi and MEMS Sensors for Indoor Navigation,” University of Calgary, 2015.

[30] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piche, “A comparative survey of WLAN location fingerprinting methods,” in *6th Workshop on Positioning, Navigation and Communication, 2009. WPNC 2009*, 2009, pp. 243–251.

[31] A. Farshad, J. Li, M. K. Marina, and F. J. Garcia, “A microscopic look at WiFi fingerprinting for indoor mobile phone localization in diverse environments,” in *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013, pp. 1–10.

[32] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in *IEEE INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, 2000, vol. 2, pp. 775–784 vol.2.

[33] M. Youssef and A. Agrawala, “The Horus WLAN location determination system,” in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, 2005, pp. 205–218.

[34] F. Seco, A. R. Jimenez, C. Prieto, J. Roa, and K. Koutsou, “A survey of mathematical methods for indoor localization,” in *IEEE International Symposium on Intelligent Signal Processing, 2009. WISP 2009*, 2009, pp. 9–14.

[35] Y. Fang *et al.*, “Application of an Improved K Nearest Neighbor Algorithm in WiFi Indoor Positioning,” in *China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III*, J. Sun, J. Liu, S. Fan, and X. Lu, Eds. Springer Berlin Heidelberg, 2015, pp. 517–524.

[36] L. B. D. Mundo, R. L. D. Ansay, C. A. M. Festin, and R. M. Ocampo, “A comparison of Wireless Fidelity (Wi-Fi) fingerprinting techniques,” in *ICTC 2011*, 2011, pp. 20–25.

[37] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[38] N. Sirola, “Mathematical Methods for Personal Positioning and Navigation,” 2007.

[39] J. Pearl, *Causality: Models, Reasoning, and Inference*. New York, US: Cambridge University Press, 2009.

[40] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz, “A Probabilistic Room Location Service for Wireless Networked Environments,” in *Proceedings of the 3rd International Conference on Ubiquitous Computing*, London, UK, UK, 2001, pp. 18–34.

[41] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, “A Probabilistic Approach to WLAN User Location Estimation,” *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, Jul. 2002.

[42] M. A. Youssef, A. Agrawala, and A. U. Shankar, “WLAN location determination via clustering and probability distributions,” in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003)*, 2003, pp. 143–150.

[43] S. Ray, W. Lai, and I. C. Paschalidis, “Statistical Location Detection with Sensor Networks,” *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2670–2683, Jun. 2006.

[44] I. C. Paschalidis and D. Guo, “Robust and distributed localization in sensor networks,” in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 933–938.

[45] I. C. Paschalidis, K. Li, and D. Guo, “Model-Free Probabilistic Localization of Wireless Sensor Network Nodes in Indoor Environments,” in *Mobile Entity Localization and Tracking in GPS-less Environments*, R. Fuller and X. D. Koutsoukos, Eds. Springer Berlin Heidelberg, 2009, pp. 66–78.

[46] I. C. Paschalidis and D. Guo, “Robust and Distributed Stochastic Localization in Sensor Networks: Theory and Experimental Results,” *ACM Trans. Sen. Netw.*, vol. 5, no. 4, pp. 34:1–34:22, Nov. 2009.

[47] D. Milioris, G. Tzagkarakis, A. Papakonstantinou, M. Papadopouli, and P. Tsakalides, “Low-dimensional signal-strength fingerprint-based positioning in wireless LANs,” *Ad Hoc Networks*, vol. 12, pp. 100–114, Jan. 2014.

[48] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, San Francisco, CA, USA, 2001, pp. 282–289.

[49] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, “Lightweight map matching for indoor localisation using conditional random fields,” in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, 2014, pp. 131–142.

[50] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second. New York, NY: Springer New York, 2009.

[51] C. Nerguizian, C. Despins, and S. Affès, “Indoor Geolocation with Received Signal Strength Fingerprinting Technique and Neural Networks,” in *Telecommunications and Networking - ICT 2004*, 2004, pp. 866–875.

[52] M. Borenovic, A. Neskovic, D. Budimir, and L. Zezelj, “Utilizing artificial neural networks for WLAN positioning,” in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2008, pp. 1–5.

[53] C. Laoudias, D. G. Eliades, P. Kemppi, C. G. Panayiotou, and M. M. Polycarpou, “Indoor Localization Using Neural Networks with Location Fingerprints,” in *Artificial Neural Networks – ICANN 2009*, 2009, pp. 954–963.

[54] M. Altini, D. Brunelli, E. Farella, and L. Benini, “Bluetooth indoor localization with multiple neural networks,” in *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, 2010, pp. 295–300.

[55] T. Pulkkinen, T. Roos, and P. Myllymäki, “Semi-supervised Learning for WLAN Positioning,” in *Artificial Neural Networks and Machine Learning – ICANN 2011*, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Springer Berlin Heidelberg, 2011, pp. 355–362.

[56] S. Thrun, “Learning metric-topological maps for indoor mobile robot navigation,” *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, Feb. 1998.

[57] D. V. Le, N. Meratnia, and P. J. M. Havinga, “Unsupervised Deep Feature Learning to Reduce the Collection of Fingerprints for Indoor Localization Using Deep Belief Networks,” in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–7.

[58] A. Haider, Y. Wei, S. Liu, and S.-H. Hwang, “Pre- and Post-Processing Algorithms with Deep Learning Classifier for Wi-Fi Fingerprint-Based Indoor Positioning,” *Electronics*, vol. 8, no. 2, p. 195, Feb. 2019.

[59] L. Wu, C.-H. Chen, and Q. Zhang, “A Mobile Positioning Method Based on Deep Learning Techniques,” *Electronics*, vol. 8, no. 1, p. 59, Jan. 2019.

[60] C. Zhang, P. Patras, and H. Haddadi, “Deep Learning in Mobile and Wireless Networking: A Survey,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.

[61] B. Ferris, D. Hähnel, and D. Fox, “Gaussian Processes for Signal Strength-Based Location Estimation,” in *Proc. of Robotics Science and Systems*, 2006.

[62] M. Jadaliha and J. Choi, “Fully Bayesian simultaneous localization and spatial prediction using Gaussian Markov random fields (GMRFs),” in *2013 American Control Conference*, 2013, pp. 4592–4597.

[63] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, Mass: MIT Press, 2006.

[64] Y. Xu, J. Choi, S. Dass, and T. Maiti, “Efficient Bayesian spatial prediction with mobile sensor networks using Gaussian Markov random fields,” *Automatica*, vol. 49, no. 12, pp. 3520–3530, Dec. 2013.

[65] D. Wang, K. O’Keefe, and M. G. Petovello, “Decentralized cooperative navigation for vehicle-to-vehicle (V2V) applications using GPS integrated with UWB range,” in *Proceedings of the ION Pacific PNT 2013 Meeting*, Honolulu, Hawaii, 2013, pp. 793–803.

[66] J. Rantakokko *et al.*, “Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization,” *Wireless Communications, IEEE*, vol. 18, no. 2, pp. 10–18, 2011.

[67] F. Olsson, J. Rantakokko, and J. Nygårds, “Cooperative Localization Using a Foot-mounted Inertial Navigation System and Ultrawideband Ranging,” in *International Conference on Indoor Positioning and Indoor Navigation*, 2014, vol. 27, p. 30th.

[68] P. Strömback *et al.*, “Foot-mounted inertial navigation and cooperative sensor fusion for indoor positioning,” in *Proceedings of the 2010 International Technical Meeting of the Institute of Navigation*, San Diego, CA, 2010, pp. 89–98.

[69] F. Berefelt, B. Boberg, J. Nygårds, P. Strömbäck, and S. Wirkander, “Collaborative GPS/INS navigation in urban environment,” in *Proceedings of the 2004 National Technical Meeting of the Institute of Navigation*, San Diego, CA, 2004, pp. 1114–1125.

[70] C. Yang and A. Soloviev, “Covariance Analysis of Spatial and Temporal Effects of Collaborative Navigation,” *Navigation*, vol. 61, no. 3, pp. 213–225, 2014.

[71] J.-O. Nilsson, D. Zachariah, I. Skog, and P. Händel, “Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging,” *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, pp. 1–17, 2013.

[72] D. Zachariah, I. Skog, M. Jansson, and P. Handel, “Bayesian Estimation with Distance Bounds,” *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 880–883, Dec. 2012.

[73] S. Tang, N. Kubo, N. Kawanishi, R. Furukawa, A. Hasegawa, and Y. Takeuchi, “Cooperative Relative Positioning for Intelligent Transportation System,” *International Journal of Intelligent Transportation Systems Research*, vol. 13, no. 3, pp. 131–142, Sep. 2015.

[74] S. Tang, N. Kubo, and M. Ohashi, “Cooperative relative positioning for intelligent transportation system,” in *ITS Telecommunications (ITST), 2012 12th International Conference on*, 2012, pp. 506–511.

[75] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54–69, 2005.

[76] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative Localization in Wireless Networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[77] M. Youssef and N. El-Sheimy, "Robust cooperative localization technique for wireless sensor networks," in *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, 2009, pp. 1–4.

[78] M. Z. Win *et al.*, "Network localization and navigation via cooperation," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 56–62, May 2011.

[79] H. L. Van Trees and K. L. Bell, *Detection Estimation and Modulation Theory: Part 1 - Detection, Estimation, and Filtering Theory*, 2nd ed. Somerset, US: Wiley, 2013.

[80] P.-H. Tseng, Z. Ding, and K.-T. Feng, "Cooperative Self-Navigation in a Mixed LOS and NLOS Environment," *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 350–363, Feb. 2014.

[81] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[82] N. A. Alsindi, K. Pahlavan, B. Alavi, and X. Li, "A Novel Cooperative Localization Algorithm for Indoor Sensor Networks," in *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2006, pp. 1–6.

[83] H. Liu *et al.*, “Push the Limit of WiFi Based Localization for Smartphones,” in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2012, pp. 305–316.

[84] Z. Yang, X. Feng, and Q. Zhang, “Adometer: Push the Limit of Pedestrian Indoor Localization through Cooperation,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2473–2483, Nov. 2014.

[85] Y. T. Li, G. Chen, and M. T. Sun, “An Indoor Collaborative Pedestrian Dead Reckoning System,” in *2013 42nd International Conference on Parallel Processing*, 2013, pp. 923–930.

[86] J. Jun *et al.*, “Social-Loc: Improving Indoor Localization with Social Sensing,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2013, pp. 14:1–14:14.

[87] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, “BeepBeep: A High Accuracy Acoustic Ranging System Using COTS Mobile Devices,” in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2007, pp. 1–14.

[88] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, Mass: MIT Press, 2005.

[89] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part I,” *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.

[90] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): Part II,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[91] M. Angermann and P. Robertson, “FootSLAM: Pedestrian Simultaneous Localization and Mapping Without Exteroceptive Sensors—Hitchhiking on Human Perception and

Cognition," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1840–1848, May 2012.

[92] J. Tan, X. Fan, S. Wang, and Y. Ren, "Optimization-Based Wi-Fi Radio Map Construction for Indoor Positioning Using Only Smart Phones," *Sensors*, vol. 18, no. 9, p. 3095, Sep. 2018.

[93] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM Using Gaussian Process Latent Variable Models," in *IJCAI*, 2007, vol. 7, pp. 2480–2485.

[94] L. Bruno and P. Robertson, "Wislam: Improving footslam with wifi," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, 2011, pp. 1–10.

[95] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor wifi graphslam," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1038–1043.

[96] S. W. Yang, S. X. Yang, and L. Yang, "Method of improving WiFi SLAM based on spatial and temporal coherence," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1991–1996.

[97] M. Zhou, Q. Zhang, K. Xu, Z. Tian, and W. He, "EDGES: Improving WLAN SLAM with Logic Graph Construction and Mapping," in *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.

[98] F. Herranz, A. Llamazares, E. Molinos, M. Ocaña, and M. A. Sotelo, "WiFi SLAM algorithms: an experimental comparison," *Robotica*, vol. 34, no. 04, pp. 837–858, Apr. 2016.

[99] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, "Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task," in *2011 15th International Conference on Advanced Robotics (ICAR)*, 2011, pp. 198–203.

[100] A. Kemppainen, I. Vallivaara, and J. Röning, “Magnetic field SLAM exploration: Frequency domain Gaussian processes and informative route planning,” in *2015 European Conference on Mobile Robots (ECMR)*, 2015, pp. 1–7.

[101] R. M. Faragher, C. Sarno, and M. Newman, “Opportunistic radio SLAM for indoor navigation using smartphone sensors,” in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, 2012, pp. 120–128.

[102] M. Hardegger, D. Roggen, S. Mazilu, and G. Tröster, “ActionSLAM: Using location-related actions as landmarks in pedestrian SLAM,” in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012, pp. 1–10.

[103] M. Hardegger, D. Roggen, and G. Tröster, “3D ActionSLAM: wearable person tracking in multi-floor environments,” *Personal and Ubiquitous Computing*, vol. 19, no. 1, pp. 123–141, Jan. 2015.

[104] P. Newman and J. Leonard, “Pure range-only sub-sea SLAM,” in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA ’03*, 2003, vol. 2, pp. 1921–1926 vol.2.

[105] N. D. Lawrence, “Gaussian process latent variable models for visualisation of high dimensional data,” *Advances in neural information processing systems*, vol. 16, no. 3, pp. 329–336, 2004.

[106] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, “Voronoi tracking: location estimation using sparse and noisy sensor data,” in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings*, 2003, vol. 1, pp. 723–728 vol.1.

[107] R. Faragher and R. Harle, “SmartSLAM—an efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing,” in *ION GNSS*, 2013, vol. 13, pp. 1–14.

[108] S. Thrun and M. Montemerlo, “The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures,” *The International Journal of Robotics Research*, vol. 25, no. 5–6, pp. 403–429, May 2006.

[109] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, “Simultaneous localization and mapping using ambient magnetic field,” in *2010 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2010, pp. 14–19.

[110] A. Dammann, R. Raulefs, and S. Zhang, “On prospects of positioning in 5G,” in *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 1207–1213.

[111] P. Zhang, J. Lu, Y. Wang, and Q. Wang, “Cooperative localization in 5G networks: A survey,” *ICT Express*, vol. 3, no. 1, pp. 27–32, Mar. 2017.

[112] E. Staudinger, M. Walter, and A. Dammann, “The 5G Localization Waveform Ranging Accuracy over Time-Dispersive Channels–An Evaluation,” in *ION GNSS+ 2016*, Portland, OR, 2016.

[113] J. A. del Peral-Rosado, J. A. López-Salcedo, S. Kim, and G. Seco-Granados, “Feasibility study of 5G-based localization for assisted driving,” in *2016 International Conference on Localization and GNSS (ICL-GNSS)*, 2016, pp. 1–6.

[114] Y. Liu, X. Shi, S. He, and Z. Shi, “Prospective Positioning Architecture and Technologies in 5G Networks,” *IEEE Network*, vol. 31, no. 6, pp. 115–121, Nov. 2017.

[115] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, “Survey of Cellular Mobile Radio Localization Methods: From 1G to 5G,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1124–1148, Second-quarter 2018.

[116] J. D. Roth, M. Tummala, and J. C. McEachen, “Fundamental Implications for Location Accuracy in Ultra-Dense 5G Cellular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1784–1795, Feb. 2019.

[117] F. Perez-Cruz, S. V. Vaerenbergh, J. J. Murillo-Fuentes, M. Lazaro-Gredilla, and I. Santamaria, “Gaussian Processes for Nonlinear Signal Processing: An Overview of Recent Advances,” *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 40–50, Jul. 2013.

[118] C. E. Rasmussen, “Gaussian processes for machine learning,” 2006.

[119] M. M. Atia, A. Noureldin, and M. J. Korenberg, “Dynamic Online-Calibrated Radio Maps for Indoor Positioning in Wireless Local Area Networks,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1774–1787, Sep. 2013.

[120] M. Aravecchia and S. Messelodi, “Gaussian process for RSS-based localisation,” in *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2014, pp. 654–659.

[121] M. Dashti, S. Yiu, S. Yousefi, F. Perez-Cruz, and H. Claussen, “RSSI Localization with Gaussian Processes and Tracking,” in *2015 IEEE Globecom Workshops (GC Wkshps)*, 2015, pp. 1–6.

[122] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed. New York: Springer, 2006.

[123] Naser El-Sheimy, “ENGO 623 - Inertial Surveying and INS/GPS Integration, Course Notes. University of Calgary, Calgary, AB,” presented at the Course Notes, University of Calgary, Calgary, AB, Winter-2016.

[124] A. Noureldin, T. B. Karamat, and J. Georgy, *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[125] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. Institution of Engineering and Technology, 2004.

[126] Y. Zhuang, H. Lan, Y. Li, and N. El-Sheimy, "PDR/INS/WiFi Integration Based on Handheld Devices for Indoor Pedestrian Navigation," *Micromachines*, vol. 6, no. 6, pp. 793–812, Jun. 2015.

[127] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," *Analog Devices AN-602 application note*, vol. 2, no. 2, pp. 1–6, 2002.

[128] A. Gelb, Joseph F. Kasper, Jr., Raymond A. Nash, Jr., Charles F. Price, and Arthur A. Sutherland, Jr., *Applied optimal estimation*. Cambridge, Mass: M.I.T. Press, 1974.

[129] D. Simon, *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*. Hoboken, N.J: Wiley-Interscience, 2006.

[130] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 2000, pp. 153–158.

[131] R. Kandepu, B. Foss, and L. Imsland, "Applying the unscented Kalman filter for nonlinear state estimation," *Journal of Process Control*, vol. 18, no. 7–8, pp. 753–768, Aug. 2008.

[132] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, Jul. 2010.

[133] V. V. Williams, “Multiplying Matrices Faster Than Coppersmith-Winograd,” in *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 2012, pp. 887–898.

[134] E. Snelson and Z. Ghahramani, “Sparse Gaussian processes using pseudo-inputs,” in *Advances in neural information processing systems*, 2005, pp. 1257–1264.

[135] J. Quiñonero-Candela and C. E. Rasmussen, “A unifying view of sparse approximate Gaussian process regression,” *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[136] K. Chalupka, C. K. I. Williams, and I. Murray, “A Framework for Evaluating Approximation Methods for Gaussian Process Regression,” *Journal of Machine Learning Research*, vol. 14, no. Feb, pp. 333–350, 2013.

[137] J. Schreiter, D. Nguyen-Tuong, and M. Toussaint, “Efficient sparsification for Gaussian process regression,” *Neurocomputing*, vol. 192, pp. 29–37, Jun. 2016.

[138] F. Yin and F. Gunnarsson, “Distributed Recursive Gaussian Processes for RSS Map Applied to Target Tracking,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 492–503, Apr. 2017.

[139] M.-G. D. Benedetto, *UWB Communication Systems: A Comprehensive Overview*. Hindawi Publishing Corporation, 2006.

[140] S. Emami, *UWB Communication Systems: Conventional and 60 GHz*. New York, NY: Springer New York, 2013.

[141] C. L. Bennett and G. F. Ross, “Time-domain electromagnetics and its applications,” *Proceedings of the IEEE*, vol. 66, no. 3, pp. 299–318, Mar. 1978.

[142] M. Z. Win and R. A. Scholtz, “On the robustness of ultra-wide bandwidth signals in dense multipath environments,” *IEEE Communications Letters*, vol. 2, no. 2, pp. 51–53, Feb. 1998.

[143] Federal Communications Commission (FCC), “New Public Safety Applications and Broadband Internet Access Among Uses Envisioned by FCC Authorization of Ultra-Wideband Technology,” 14-Feb-2002.

[144] X. Gu and L. Taylor, “Ultra-Wideband and Its Capabilities,” *BT Technology Journal*, vol. 21, no. 3, pp. 56–66, Jul. 2003.

[145] W. Hirt, “The European UWB Radio Regulatory and Standards Framework: Overview and Implications,” in *2007 IEEE International Conference on Ultra-Wideband*, 2007, pp. 733–738.

[146] “IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),” *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, Sep. 2011.

[147] J. Zhang, P. V. Orlik, Z. Sahinoglu, A. F. Molisch, and P. Kinney, “UWB Systems for Wireless Sensor Networks,” *Proceedings of the IEEE*, vol. 97, no. 2, pp. 313–331, Feb. 2009.

[148] S. Gezici *et al.*, “Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks,” *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, Jul. 2005.

[149] F. Mazhar, M. G. Khan, and B. Sällberg, “Precise Indoor Positioning Using UWB: A Review of Methods, Algorithms and Implementations,” *Wireless Pers Commun*, vol. 97, no. 3, pp. 4467–4491, Dec. 2017.

[150] A. F. G. Ferreira, D. M. A. Fernandes, A. P. Catarino, and J. L. Monteiro, "Localization and Positioning Systems for Emergency Responders: A Survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2836–2870, Fourth-quarter 2017.

[151] Z. Sahinoglu and S. Gezici, "Ranging in the IEEE 802.15.4a Standard," in *2006 IEEE Annual Wireless and Microwave Technology Conference*, 2006, pp. 1–5.

[152] T. Gigl, G. J. M. Janssen, V. Dizdarevic, K. Witrisal, and Z. Irahhauten, "Analysis of a UWB Indoor Positioning System Based on Received Signal Strength," in *Navigation and Communication 2007 4th Workshop on Positioning*, 2007, pp. 97–101.

[153] Y. Qi and H. Kobayashi, "On relation among time delay and signal strength based geolocation methods," in *GLOBECOM '03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489)*, 2003, vol. 7, pp. 4079–4083 vol.7.

[154] A. E. Waadt *et al.*, "Positioning in multiband OFDM UWB utilizing received signal strength," in *Navigation and Communication 2010 7th Workshop on Positioning*, 2010, pp. 308–312.

[155] S. Wang *et al.*, "System implementation study on RSSI based positioning in UWB networks," in *2010 7th International Symposium on Wireless Communication Systems*, 2010, pp. 36–40.

[156] Decawave, "DW1000 User Manual," Decawave Ltd., 2016.

[157] Pozyx Labs, "Pozyx - Accurate Positioning," 2018. [Online]. Available: https://www.pozyx.io. [Accessed: 15-May-2018].

[158] A. Masiero, F. Fissore, A. Guarnieri, F. Pirotti, D. Visintini, and A. Vettore, "Performance Evaluation of Two Indoor Mapping Systems: Low-Cost UWB-Aided Photogrammetry and Backpack Laser Scanning," *Applied Sciences*, vol. 8, no. 3, p. 416, Mar. 2018.

[159] “Creator - Python - Multitag positioning,” *Pozyx NV*. [Online]. Available: https://www.pozyx.io/documentation/creator/python/multitag-positioning. [Accessed: 26-Feb-2019].

[160] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, “MAC Essentials for Wireless Sensor Networks,” *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 222–248, Second 2010.

[161] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, “The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 101–120, First 2013.

[162] M. Doudou, D. Djenouri, and N. Badache, “Survey on Latency Issues of Asynchronous MAC Protocols in Delay-Sensitive Wireless Sensor Networks,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 528–550, Second 2013.

[163] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, and N. M. Khan, “A Critical Analysis of Research Potential, Challenges, and Future Directives in Industrial Wireless Sensor Networks,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 39–95, First-quarter 2018.

[164] V. Quintero, C. Estevez, M. Orchard, and A. Pérez, “Improvements of Energy-Efficient Techniques in WSNs: A MAC-Protocol Approach,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[165] G. Franceschetti, S. Stornelli, and S. Stornelli, *Wireless Networks: From the Physical Layer to Communication, Computing, Sensing and Control*. San Diego, UNITED STATES: Elsevier Science & Technology, 2006.

[166] E. Karapistoli, F.-N. Pavlidou, I. Gragopoulos, and I. Tsetsinas, “An overview of the IEEE 802.15.4a Standard,” *IEEE Communications Magazine*, vol. 48, no. 1, pp. 47–53, Jan. 2010.

[167] Decawave, “DWM1000 Datasheet,” Decawave Ltd., Datasheet, 2016.

[168] TI, “LAUNCHXL-CC2640R2 SimpleLink™ Bluetooth® low energy CC2640R2F wireless MCU LaunchPad™ development kit | TI.com.” [Online]. Available: http://www.ti.com/tool/LAUNCHXL-CC2640R2. [Accessed: 27-Feb-2019].

[169] A. R. J. Ruiz and F. S. Granja, “Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 2106–2117, Aug. 2017.

[170] L. G. Roberts, “ALOHA Packet System with and Without Slots and Capture,” *SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, Apr. 1975.

[171] A. Rajandekar and B. Sikdar, “A Survey of MAC Layer Issues and Protocols for Machine-to-Machine Communications,” *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 175–186, Apr. 2015.

[172] M. Ridolfi, S. Van de Velde, H. Steendam, and E. De Poorter, “Analysis of the Scalability of UWB Indoor Localization Solutions for High User Densities,” *Sensors*, vol. 18, no. 6, p. 1875, Jun. 2018.

[173] M. Sakr and N. El-Sheimy, “Efficient Wi-Fi Signal Strength Maps Using Sparse Gaussian Process Models,” in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo, Japan, 2017, pp. 1–8.

[174] Tom Lankhorst, “DecaWave DW1000 TX Frame Time Estimation,” 13-Apr-2016. [Online]. Available: https://tomlankhorst.nl/estimating-decawave-dw1000-tx-time/. [Accessed: 06-Mar-2019].

[175] Decawave, “Sources of Error in DW1000 Based Two-Way Ranging (TWR) Schemes,” Decawave Ltd., Application Note APS011, 2014.

[176] C. Yang and A. Soloviev, “Distributed Estimation for Vehicular Collaborative Navigation,” in *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, Tampa, Florida, 2014, pp. 211–222.

[177] D. Woodbury and J. Junkins, “On the Consider Kalman Filter,” in *AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario, Canada, 2010.

[178] M. Sakr, A. Masiero, and N. El-Sheimy, “Evaluation of Dynamic Ad-Hoc UWB Indoor Positioning System,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII–1, pp. 379–385, Sep. 2018.

[179] M. Sakr and N. El-Sheimy, “Performance Analysis of a New Collaborative and Multimodal Framework for Indoors Navigation using Smartphones,” in *ION GNSS+ 2016*, Portland, OR, 2016, pp. 1114–1124.

[180] M. Sakr and N. El-Sheimy, “Filter Architectures for Indoor Localization Using Gaussian Process on Smartphones,” in *Proceedings of the 10th International Conference on Mobile Mapping Technology (MMT 2017)*, Cairo, Egypt, 2017.

[181] C. Yu, N. El-Sheimy, H. Lan, and Z. Liu, “Map-Based Indoor Pedestrian Navigation Using an Auxiliary Particle Filter,” *Micromachines*, vol. 8, no. 7, p. 225, Jul. 2017.

[182] O. Hlinka, F. Hlawatsch, and P. M. Djuric, “Distributed particle filtering in agent networks: A survey, classification, and comparison,” *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, Jan. 2013.

[183] W. Li, Z. Wang, Y. Yuan, and L. Guo, “Particle filtering with applications in networked systems: a survey,” *Complex Intell. Syst.*, vol. 2, no. 4, pp. 293–315, Dec. 2016.

[184] S. Grime and H. F. Durrant-Whyte, “Data fusion in decentralized sensor networks,” *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, Oct. 1994.

[185] M. Rosencrantz, G. Gordon, and S. Thrun, “Decentralized Sensor Fusion with Distributed Particle Filters,” in *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, 2003, pp. 493–500.

[186] M. Coates, “Distributed Particle Filters for Sensor Networks,” in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, New York, NY, USA, 2004, pp. 99–107.

[187] J. Read, K. Achutegui, and J. Míguez, “A distributed particle filter for nonlinear tracking in wireless sensor networks,” *Signal Processing*, vol. 98, pp. 121–134, May 2014.

[188] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch, “Distributed Localization and Tracking of Mobile Networks Including Noncooperative Objects,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 1, pp. 57–71, Mar. 2016.

[189] K. Kang, V. Maroulas, I. Schizas, and F. Bao, “Improved distributed particle filters for tracking in a wireless sensor network,” *Computational Statistics & Data Analysis*, vol. 117, pp. 90–108, Jan. 2018.

[190] B. Wang and Y. Tian, “Distributed Network Localization: Accurate Estimation With Noisy Measurement and Communication Information,” *IEEE Transactions on Signal Processing*, vol. 66, no. 22, pp. 5927–5940, Nov. 2018.

[191] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, “Nonparametric belief propagation for self-localization of sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[192] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, “Nonparametric Belief Propagation,” *Commun. ACM*, vol. 53, no. 10, pp. 95–103, Oct. 2010.

[193] Z. Lin, T. Han, R. Zheng, and M. Fu, “Distributed Localization for 2-D Sensor Networks with Bearing-Only Measurements Under Switching Topologies,” *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6345–6359, Dec. 2016.

[194] S. Tomic, M. Beko, R. Dinis, and P. Montezuma, “Distributed algorithm for target localization in wireless sensor networks using RSS and AoA measurements,” *Pervasive and Mobile Computing*, vol. 37, pp. 63–77, Jun. 2017.

[195] S. Safavi, U. A. Khan, S. Kar, and J. M. F. Moura, “Distributed Localization: A Linear Theory,” *Proc. IEEE*, pp. 1–20, 2018.

[196] A. Carron, M. Todescato, R. Carli, and L. Schenato, “An Asynchronous Consensus-Based Algorithm for Estimation From Noisy Relative Measurements,” *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 283–295, Sep. 2014.

[197] P. Barooah and J. P. Hespanha, “Estimation on graphs from relative measurements,” *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 57–74, Aug. 2007.

[198] P. Barooah and J. P. Hespanha, “Estimation From Relative Measurements: Electrical Analogy and Large Graphs,” *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2181–2193, Jun. 2008.

[199] W. S. Rossi, P. Frasca, and F. Fagnani, “Distributed Estimation from Relative and Absolute Measurements,” *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6385–6391, Dec. 2017.

[200] C. Ravazzi, N. Chan, and P. Frasca, “Distributed estimation from relative measurements of heterogeneous and uncertain quality,” *IEEE Transactions on Signal and Information Processing over Networks*, pp. 1–1, 2018.

[201] N. M. Freris and A. Zouzias, “Fast distributed smoothing of relative measurements,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 1411–1416.

[202] T. Strohmer and R. Vershynin, “A Randomized Kaczmarz Algorithm with Exponential Convergence,” *J Fourier Anal Appl*, vol. 15, no. 2, p. 262, Apr. 2008.

[203] M. Todescato, A. Carron, R. Carli, A. Franchi, and L. Schenato, “Multi-robot localization via GPS and relative measurements in the presence of asynchronous and lossy communication,” in *2016 European Control Conference (ECC)*, 2016, pp. 2527–2532.

[204] P. Frasca, H. Ishii, C. Ravazzi, and R. Tempo, “Distributed randomized algorithms for opinion formation, centrality computation and power systems estimation: A tutorial overview,” *European Journal of Control*, vol. 24, pp. 2–13, Jul. 2015.

[205] C. Ravazzi, P. Frasca, R. Tempo, and H. Ishii, “Ergodic Randomized Algorithms and Dynamics Over Networks,” *IEEE Transactions on Control of Network Systems*, vol. 2, no. 1, pp. 78–87, Mar. 2015.

[206] T. Schön, F. Gustafsson, and P. J. Nordlund, “Marginalized particle filters for mixed linear/nonlinear state-space models,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[207] T. B. Schön, R. Karlsson, and F. Gustafsson, “The marginalized particle filter – analysis, applications and generalizations,” *ESAIM: Proceedings*, vol. 19, pp. 53–64, 2007.

[208] F. Daum and J. Huang, “Curse of dimensionality and particle filters,” in *2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652)*, 2003, vol. 4, pp. 4_1979-4_1993.

[209] S. Särkkä, *Bayesian Filtering and Smoothing*. New York, NY, USA: Cambridge University Press, 2013.

[210] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, Jul. 2000.

[211] S. F. Schmidt, “Application of State-Space Methods to Navigation Problems,” in *Advances in Control Systems*, vol. 3, C. T. Leondes, Ed. New York: Academic Press, 1966, pp. 293–340.

[212] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2003.

[213] H. L. Van Trees and K. L. Bell, Eds., *Bayesian bounds for parameter estimation and nonlinear filtering/tracking*. Hoboken, NJ: IEEE Press, 2007.

[214] P. Tichavsky, C. H. Muravchik, and A. Nehorai, “Posterior Cramer-Rao bounds for discrete-time nonlinear filtering,” *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, May 1998.

[215] E. G. Larsson, “Cramer-Rao bound analysis of distributed positioning in sensor networks,” *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 334–337, Mar. 2004.

[216] J. Chaffee and J. Abel, “GDOP and the Cramer-Rao bound,” in *Proceedings of 1994 IEEE Position, Location and Navigation Symposium - PLANS’94*, 1994, pp. 663–668.

[217] A. Dogandzic and A. Nehorai, “Cramer-Rao bounds for estimating range, velocity, and direction with an active array,” *IEEE Transactions on Signal Processing*, vol. 49, no. 6, pp. 1122–1137, Jun. 2001.

[218] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[219] B. Z. Bobrovsky, E. Mayer-Wolf, and M. Zakai, “Some Classes of Global Cramer-Rao Bounds,” *The Annals of Statistics*, vol. 15, no. 4, pp. 1421–1438, 1987.

[220] C. Hue and P. P. Rez, “Posterior Cramer-Rao Bounds for Multi-Target Tracking,” vol. 42, no. 1, p. 13, 2006.

[221] C. Fritsche, E. Özkan, L. Svensson, and F. Gustafsson, “A fresh look at Bayesian Cramér-Rao bounds for discrete-time nonlinear filtering,” in *17th International Conference on Information Fusion (FUSION)*, 2014, pp. 1–8.

[222] J. Torres-Sospedra *et al.*, “Off-Line Evaluation of Mobile-Centric Indoor Positioning Systems: The Experiences from the 2017 IPIN Competition,” *Sensors*, vol. 18, no. 2, p. 487, Feb. 2018.

[223] “IPIN 2018.” [Online]. Available: http://ipin2018.ifsttar.fr/. [Accessed: 02-Jun-2019].

[224] T. B. Schon, F. Gustafsson, and R. Karlsson, “The Particle Filter in Practice,” in *The Oxford Handbook of Nonlinear Filtering*, Oxford, NY: Oxford University Press, 2011, pp. 741–767.